RoCHI - INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION

Proceedings of the 17th International Conference on Human–Computer Interaction - RoCHI 2020, 22–23 October, Sibiu, Romania

Edited by:

Teodor Ștefănuț
Technical University of Cluj-Napoca, Romania

Jean Vanderdonckt
Université catholique de Louvain, Belgium

Alex Butean
Lucian Blaga University of Sibiu, Romania

MATRIX ROM
Bucharest, 2020

Publisher

MATRIX ROM

C.P. 16-162

062510 – Bucureşti, România

Tel.: 021 4113617, Fax: 021 4114280

E-mail: office@matrixrom.ro

www.matrixrom.ro

MatrixRom publishing house is a certified publisher by the National Council for Scientific Research in Higher Education (Consiliul Naţional al Cercetării Ştiinţifice din Învăţământul Superior)

Cover graphic design: Sabin-Corneliu Buraga, *A.I.Cuza University, Iasi*

Graphic identity: Adrian Mironescu, *Idegrafo*

RoCHI – International Conference on Human Computer Interaction (2020)

http://rochi.utcluj.ro/proceedings/en/index.php

ISSN 2501-9422
ISSN-L 2501-9422

CONTENTS

Invited Keynote Paper

Interaction and Digital Humanities

Towards a more respectful society

User interfaces in multiple contexts of use

Developing the User Interface

Conversational agents

Interacting by voice and chatbot

Virtual/Augmented/Mixed Reality forever

Classroom HCI

ORGANISATION

Conference Chair

Alex Butean, *Lucian Blaga University of Sibiu, Romania*

Program Committee Chair

Jean Vanderdonckt, *Université catholique de Louvain, Belgium*

Organizing Committee Chair

Volovici Daniel, *Lucian Blaga University of Sibiu, Romania*

RoCHI Conference Website Administrator

Teodor Ștefănuț, *Technical University of Cluj-Napoca, Romania*

Associate Chairs

Dorian Gorgan, *Technical University of Cluj-Napoca, Romania*
Adrian Iftene, *Alexandru Ioan Cuza University, Iaşi, Romania*
Costin Pribeanu, *Academy of Romanian Scientists, Romania*
Ștefan Trăușan-Matu, *University Politehnica of Bucharest, Romania*

Organizing Committee

Constantin Zamfirescu, *Lucian Blaga University of Sibiu, Romania*
Adrian Florea, *Lucian Blaga University of Sibiu, Romania*
Remus Brad, *Lucian Blaga University of Sibiu, Romania*

Program Committee

Sabin-Corneliu Buraga, *Alexandru Ioan Cuza University, Iaşi, Romania*
Grigoreta Sofia Cojocar, *Babeş-Bolyai University, Cluj-Napoca, Romania*
Marian Dârdală, *Academy of Economic Studies of Bucharest, Romania*
Mihai Dascălu, *University Politehnica of Bucharest, Romania*
Anne-Marie Pinna-Dery, *University of Nice Sophia Antipolis, Nice, France*
Philippe Dessus, *Université Grenoble Alpes, France*
Alan Dix, *Swansea University, UK*
Bruno Dumas, *Université de Namur, Belgium*
Peter Forbrig, *University of Rostock, Germany*
Vivian Genaro Motti, *George Mason University, Fairfax, United States*
Juan Gonzalez Calleros, *Benemérita Universidad Autónoma de Puebla, Mexico*
Dorian Gorgan, *Technical University Cluj-Napoca, Romania*
Adriana-Mihaela Guran, *Babeş-Bolyai University, Cluj-Napoca, Romania*
Adrian Iftene, *A.I.CuzaUniversity, Iaşi, Romania*
Dragoş Daniel Iordache, *ICI Bucharest, Romania*
Suzanne Kieffer, *Université catholique de Louvain, Belgium*
Cristophe Kolski, *Université Polytechnique Hauts-de-France, Valenciennes, France*
Marta Larusdottir, *Reykjavik University, Iceland*
Alin Moldoveanu, *University Politehnica of Bucharest, Romania*
Vivian Motti, *LIRO,Ireland*
Luis Olsina, *University Rio de laPlata, Argentina*
Philippe Palanque, *IRIT, Universite Paul Sabatier, France*
Oscar Pastor, *Polytechnic University of Valencia, Spain*
Dorin Mircea Popovici, *University Ovidius of Constanţa, Romania*
Jorge Luis PerezMedina, *Universidad de Las Americas, Ecuador*
Costin Pribeanu, *Academy of Romanian Scientists. Romania*
Traian Eugen Rebedea, *University Politehnica of Bucharest, Romania*
Adriana-Elena Reveiu, *Academy of Economic Studies of Bucharest, Romania*
Jenny Ruiz de la Pena, *Universidad de Holguín, Cuba*
Carmen Santoro, *ISTI-CNR Pisa, Italy*
Corina Sas, *Lancaster University*
Marcin Sikorski, *Gdansk University of Technology, Poland*
Christian Stary, *University of Linz*
Teodor Ştefănuţ, *Technical University of Cluj-Napoca, Romania*
Ştefan Trăuşan-Matu, *Politehnica University of Bucharest, Romania*
Jean Vanderdonckt, *Université catholique de Louvain, Belgium*

Additional Reviewers

Felix Albu, *Valahia University of Targoviste, Romania*
Diana Andone, *Politechnica Univeristy of Timișoara, Romania*
Jeferson Arango Lopez, *Universidad de Caldas, Colombia*
Victor Bâcu, *Technical University of Cluj-Napoca, Romania*
Florina-Oana Bălan, *Politechnica University of Bucharest, Romania*
Elena Băutu, *Ovidius University of Contanța, Romania*
Bianca-Cerasela-Zelia Blaga, *Technical University Cluj-Napoca, Romania*
Nicolas Burny, *Universite Catholique de Louvain, Belgium*
Alex Butean, *Lucian Blaga University of Sibiu, Romania*
Dumitru-Clementin Cercel, *Politechnica University of Bucharest, Romania*
Mihaela Colhon, *University of Craiova, Romania*
Cesar A. Collazos, *University of Cauca, Colombia*
Alin-Marius Cruceat, *Lucian Blaga University of Sibiu, Romania*
Marian Dărdală, *University of Economics, Bucharest, Romania*
Daniela Gifu, *A.I Cuza University of Iași, Romania*
Florin Girbacia, *Transilvania University of Brasov, Romania*
Gabriel Gorghiu, *Valahia University of Târgoviște, Romania*
Gabriel Gutu-Robu, *Politechnica University of Bucharest, Romania*
Marilena Ianculescu, *ICI Bucharest, Romania*
Iyad Khaddam, *Université Catholique de Louvain, Belgium*
Ketoma Vix Kemanji, *Heilbronn University, Germany*
Vincentas Lamanauskas, *Siauliai University, Lithuania*
Valentina Marinescu, *University of Bucharest, Romania*
Alexandru Matei, *Lucian Blaga University of Sibiu, Romania*
Cristian Mihăescu, *University of Craiova, Romania*
Verginica Mititelu, *RACAI Bucharest, Romania*
Delia Mitrea, *Technical University of Cluj-Napoca, Romania*
Florica Moldoveanu, *Politechnica University of Bucharest, Romania*
Gabriel Neagu, *ICI Bucharest, Romania*
Mihaela Ordean, *Gemma Computing, Romania*
Marian Pădure, *Babaes Bolyai University, Cluj-Napoca, Romania*
Elvira Popescu, *University of Craiova, Romania*
Paul Ștefan Popescu, *University of Craiova, Romania*
Dan Rotar, *Vasile Aecsandri University of Bacău, Romania*
Cristian Rusu, *Pontificia Universidad Catolica de Valparaiso, Chile*
Adrian Sabou, *Technical University Cluj-Napoca, Romania*
Emil Stănescu, *ICI Bucharest, Romania*
Radu-Daniel Vătavu, *Stefan cel Mare University of Suceava, Romania*
Silviu Vert, *Politechnica Univeristy of Timișoara, Romania*

Conference organized with the support of:

Lucian Blaga University of Sibiu,
Romania

Université Catholique de Louvain,
Belgium

RoCHI - ACM SIGCHI România

FORWORD

The International Conference on Human-Computer Interaction - RoCHI, has reached its 17th edition in 2020. RoCHI is the premier scientific forum on research and development of user interfaces and human-computer interaction in Romania providing the opportunity for exchange of ideas, expertise and research results in the field of human-computer interaction.

The invited paper authored by Professor Philippe Palanque (Université Toulouse III – Paul Sabatier, France) is entitled *User Tasks Description: a Retrospective, Recent Contributions and some Research Challenges*.

The second invited talk is entitled *Computational Modeling of Handwriting Movements* and is presented by Dr. Luis A. Leiva (Aalto University, Finland).

The 24 accepted papers to RoCHI 2020 were selected from a total of 34 submissions (70% acceptance rate) after a careful review process, each submission being assigned to a minimum of three and maximum of six reviewers.

Accepted papers have been grouped into eight sessions, each covering different subjects in Human-Computer Interaction:

- Interaction and Digital Humanities
- Towards a more respectful society
- User interfaces in multiple contexts of use
- Developing the User Interface
- Conversational agents
- Interacting by voice and chatbot
- Virtual/Augmented/Mixed Reality forever
- Classroom HCI

We would like to express our appreciation to the members of the Scientific Committee and to the volunteer reviewers who helped for selecting the best papers to be presented at the conference. Moreover, we acknowledge the efforts of all the persons involved in the organization of the RoCHI 2020 Conference and thank them for their dedication!

Finally, we would like to thank all of those who have contributed in any other way to the success of RoCHI 2020 including authors, administrators, technicians, attendees and all the involved institutions.

Editors,

Teodor Ștefănuț, Jean Vanderdonckt, and Alex Butean

User Tasks Description: a Retrospective, Recent Contributions and some Research Challenges

Philippe Palanque
ICS-IRIT, Université Toulouse
III – Paul Sabatier
118, route de Narbonne, 31042
Toulouse, France
palanque@irit.fr

DOI: 10.37789/rochi.2020.1.1.1

ABSTRACT

Describing users tasks has been the focus of research for many years starting with the seminal work from Annett and Duncan in 1967 [2]. Since then, the Human Factors and Human Computer Interaction domains have proposed multiple contributions identifying the elements that have to be gathered and represented in order to describe precisely the relevant aspects of users tasks. This keynote will highlight these fundamental elements of tasks descriptions and will state the current state of the art. A specific view on how to use such descriptions to design and assess automation will be given. Some publicly available tools will also be presented together with their use in various industrial application domains. These applications will be the opportunity to identify remaining research challenges for tasks description and modeling.

Author Keywords

Task descriptions; task modeling; task-centered design; task-centered evaluation.

ACM Classification Keywords

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; *Interactive Systems and Tools*

HISTORICAL PERSPECTIVE

Tasks Analysis and Tasks Representation are cornerstones of User Centered Design approaches, aiming to collect information from users goals, work and activities. According to Johnson [16] “any Task Analysis is comprised of three major activities; first, the collection of data; second, the analysis of that data; and third, the modelling of the task domain” (p.165). While this work emphasises the importance of data, it is important to note that users’ work is mainly a procedural activity. The means (notations and tools for representing the outcomes of task analysis) have important implications for the value and insight gained from the process, not least because any omissions cannot be discussed (among the stakeholders) or taken into consideration in later design phases.

The expressive power of the notation used to store and organize the information collected is thus a key element that is put forward by researchers proposing new notations [4]. Since the seminal HTA notation proposed by [1][2], relatively few notations to describe user tasks have been proposed and they all tend to remain unchanged after their creation. While the notations remain largely constant, their associated tools typically evolve to address new challenges. For instance, Paterno’s team has proposed several tools exploiting CTT notation since its creation in 1997 [35]: the original CTTe tool **Error! Reference source not found.** supported editing, simulating and verifying CTT task models; CTTEVis [36] added support for visualization; and a new tool added support for collaborative modelling [20]. The stability of notations contrasts with the constant evolution of application domains, technologies, and the nature of work operators’ work. This evolution can create a gap between what the notation can describe and the actual work, limiting the scope and potential benefits of using the notation at all. For instance, a notation like KMAD [4] or CTT [35] would produce the same representation of tasks for interacting with a calculator regardless of whether the calculator was a physical device, a desktop application, or an app on a mobile phone. This lack of precision and detail make it impossible for analysts to assess the interaction implications of moving from one technology to the other one.

This is heavily constrated with the work on the HAMSTERS notation that was designed after the fundamental elements of users’ tasks descriptions were identified and incorporated in notations.

HAMSTERS was built exploiting these fundamental elements that are:

- Hierarchical structure of tasks: Hierarchical Task Analysis [1] [2],
- Knowledge representation: Task Knowledge Structure [17],
- Multi-user/Collaborative activities: Groupware Task Analysis [40].

In order to address the challenges brought by new technologies, for each of the fundamentals elements above, HAMSTERS was extended to represent much more precise information:

- Tasks structuring using sub models [27] and components [10]

- Detailed knowledge and information representation [26] ;
- Synchronous, asynchronous, same place, different place collaboration [21].

More rencently, HAMSTERS notation and its associated tool HAMSTERS|XL have been released to allow human factors analysts to customize the HAMSTERS notation to the needs of the application domain or the interactive systems they are addressing. For instance, in [8] the HAMSTERS notation was extended to address the specificities of Cyber Physical Systems.

EXPLOITING TASKS DESCRIPTIONS

The following non-exhaustive list identifies some of the objectives of task analysis, highlighting its broad range of uses:

- Identification and description of the required functions for interactive system [13] [35],
- Identification and description of knowledge required to perform a task [7] [17] [26] [38],
- Identification and description of the temporal ordering of the user actions with the system [18] [22] [35],
- Identification and description of the different user roles and actors for groupware systems [37] [40],
- Identification and description of workflow between users for collaborative activities [37] [40],
- Understanding of an application domain [34],
- Recording the results of interdisciplinary discussions [30] [34],
- Production of scenarios for user evaluation [41] as well identification and generation of relevant test cases [6],
- Heuristic evaluation of usability of interactive applications [5] [37],
- Predictive assessment of task complexity and workload (motor, cognitive, perceptive) [29],
- Predictive assessment of user performance when interacting with the system [15],
- Exploration of the range of ways in which the system may be used [37],
- Preparation of training programs [1] [2] [25],
- Production of user manual [12] [33] and contextual help [14] [31] [33] [34],
- Identification and description of possible allocation of functions and tasks between the system and the user [23] [32],
- Designing new applications consistent with the user conceptual model [34],
- Identification and description of potential user errors [9] [39] [42] [24],
- Assessing other properties than usability as, for instance, dependability [43], security [44] or user experience [45].

Task analysis is thus a pillar of UCD approaches for the design of interactive systems. If the results of task analysis do not contain sufficient information, the missing information may negatively affect the design of the interactive system and its usability.

CHALLENGES AHEAD

The main challenge ahead of users' tasks representation lays in the fact that there is reluctance to use this tool that requires deep understanding and description of users' work.

Beyond, curricula like the ACM 2013 Computer Science curricula that exhibits multiple courses on Human-Computer Interaction does not even mention the need to analyse and represent users' tasks [3]. There is still a long road ahead before seing widespread use of task modeling noations and tools but the path created by CTTE which offered the first publicly available tool increased the takeup-ability of the multiple benefits of users' tasks representation.

REFERENCES

[1] John Anett. 2004. Hierarchical Task Analysis. In Diaper Dan, Stanton Neville (Eds), *The Handbook of Task Analysis for Human-Computer Interaction* (pp. 67-82). Lawrence Erlbaum Associates.

[2] John Annett, Keith Duncan.1967. Task analysis and training design. *Occupational psychology*, 41, 211-221.

[3] ACM CS curriculum 2013 https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf

[4] Sybille Caffiau, Dominique Scapin, Patrick Girard, Mickaël Baron, and Francis Jambon. 2010. Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. *Interact. with Comput.* 22, 6 (November 2010), 569-593.

[5] Cockton, G., & Woolrych, A. (2001). Understanding inspection methods: Lessons from an assessment of heuristic evaluation. *People and Computers XV, joint Proceedings of HCI 2001 and IHM 2001*, Springer Verlag 171–192.

[6] José Creissac Campos, Camille Fayollas, Marcelo Gonçalves, Célia Martinie, David Navarre, Philippe Palanque, and Miguel Pinto. 2017. A More Intelligent Test Case Generation Approach through Task Models Manipulation. Proc. ACM Hum.-Comput. Interact. 1, EICS, Article 9 (June 2017), 20 pages.

[7] Dan Diaper. 1990. Task Analysis for Knowledge Descriptions (TAKD): The Method and an Example. In *Task Analysis for Human-Computer Interaction*, D. Diaper (ed.), Ellis Horwood, pp. 108-159.

[8] R. Fahssi, C. Martinie and P. Palanque, "Embedding explicit representation of cyber-physical elements in task models," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, 2016, pp. 001969-001974, doi: 10.1109/SMC.2016.7844528.

[9] Racim Fahssi, Celia Martinie, Philippe Palanque. 2015. Enhanced Task Modelling for Systematic Identification and Explicit Representation of Human Errors. In: Abascal J., Barbosa S., Fetter M., Gross T., Palanque P., Winckler M. (eds) *IFIP TC13 Conference on Human-Computer Interaction – INTERACT 2015*. Lecture Notes in Computer Science, vol 9299.

[10] Peter Forbrig, Célia Martinie, Philippe Palanque, Marco Winckler, and Racim Fahssi. 2014. Rapid Task-Models Development Using Sub-models, Sub-routines and Generic Components. In *Proceedings of the 5th IFIP WG 13.2 International Conference on Human-Centered Software Engineering* - Volume 8742 (HCSE 2014) Vol. 8742. Springer-Verlag, 144-163.

[11] Matthias Giese, Tomasz Mistrzyk, Andreas Pfau, Gerd Szwillus, and Michael Detten. 2008. AMBOSS: A Task Modelling Approach for Safety-Critical Systems. In *Proceedings of the 2nd Conference on Human-Centered Software Engineering and 7th International Workshop on Task Models and Diagrams (HCSE-TAMODIA '08)*, Springer-Verlag, Berlin, Heidelberg, 98-109.

[12] Gong, R. & Elkerton, J. (1990). Designing minimal documentation using the GOMS model: A usability evaluation of an engineering approach. *CHI 90 Proceedings*. New York, ACM DL.

[13] Saul Greenberg. Working through Task-Centered System Design. In Diaper, D. and Stanton, N. (Eds) *The Handbook of Task Analysis for Human-Computer Interaction*. Lawrence Erlbaum Associates (2004). p49-66.

[14] Valeria Gribova. A method of Context-Sensitive Help Generation Using a Task Project. *International Journal on Information Theories & Applications* Vol.15, pp. 391-395, 2008.

[15] Bonnie E. John and David E. Kieras. 1996. The GOMS family of user interface analysis techniques: comparison and contrast. *ACM Trans. Comput.-Hum. Interact.* 3, 4 (December 1996), 320-351.

[16] Peter Johnson. 1992. Human-Computer Interaction: psychology, task analysis and software engineering, McGraw Hill, Maidenhead, UK.

[17] Peter Johnson, H. Johnson and F. Hamilton. 2000. Getting the Knowledge into HCI: Theoretical and Practical Aspects of Task Knowledge Structures. In. *Cognitive Task Analysis*. J. Schraagen, S. Chipman, V. Shalin LEA

[18] Frédéric Jourde, Yann Laurillau, and Laurence Nigay. 2010. COMM notation for specifying collaborative and multimodal interactive systems. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10)*. ACM, New York, NY, USA, 125-134.

[19] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. 2000. DENIM: finding a tighter fit between tools and practice for Web site design. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 510-517.

[20] Marco Manca , Fabio Paternò, Carmen Santoro. 2016. Collaborative Task Modelling on the Web. In: Bogdan C. et al. (eds) Human-Centered and Error-Resilient Systems Development. *HESSD 2016, HCSE 2016*. Lecture Notes in Computer Science, vol 9856. Springer,.

[21] Célia Martinie, Eric Barboni, David Navarre, Philippe Palanque, Racim Fahssi, Erwann Poupart, and Eliane Cubero-Castan. 2014. Multi-models-based engineering of collaborative systems: application to collision avoidance operations for spacecraft. In *Proceedings of the 2014 ACM SIGCHI symposium on Engineering interactive computing systems (EICS '14)*. ACM, USA, 85-94.

[22] Célia Martinie, David Navarre, Philippe Palanque, and Camille Fayollas. 2015. A generic tool-supported framework for coupling task models and interactive applications. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '15)*. ACM, New York, NY, USA, 244-253.

[23] Célia Martinie, Philippe Palanque, Eric Barboni and Martina Ragosta. 2011. Task-model based assessment of automation levels: Application to space ground segments. *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3267-3273.

[24] Célia Martinie, Philippe A. Palanque, Racim Fahssi, Jean-Paul Blanquart, Camille Fayollas, Christel Seguin. 2016. Task Model-Based Systematic Analysis of Both System Failures and Human Errors. IEEE Trans. Human-Machine Systems 46(2), 243-254.

[25] Célia Martinie, Philippe Palanque, David Navarre, Marco Winckler, and Erwann Poupart. 2011. Model-based training: an approach supporting operability of critical interactive systems. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11)*. ACM, New York, NY, USA, 53-62.

[26] Célia Martinie, Philippe Palanque, Martina Ragosta, and Racim Fahssi. 2013. Extending procedural task models by systematic explicit integration of objects, knowledge and information. In *Proceedings of the 31st European Conference on Cognitive Ergonomics (ECCE '13)*. ACM, Article 23, 10 pages.

[27] Célia Martinie, Philippe Palanque, and Marco Winckler. 2011. Structuring and composition mechanisms to address scalability issues in task models. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III (INTERACT'11)*, Pedro Campos, Nuno Nunes, Nicholas Graham, Joaquim Jorge, and Philippe Palanque (Eds.), Vol. Part III. Springer-Verlag, Berlin, Heidelberg, 589-609.

[28] Célia Martinie, Philippe Palanque, Elodie Bouzekri, Andy Cockburn, Alexandre Canny, and Eric Barboni. 2019. Analysing and Demonstrating Tool-Supported Customizable Task Notations. *Proc. ACM Hum.-Comput. Interact.* 3, EICS, Article 12 (June 2019), 26 pages. DOI:https://doi.org/10.1145/3331154

[29] O'Donnell, R. D.; Eggemeier, F. T. Workload Assessment Methodology; In K. R. Boff & L. Kaufman & J. P. Thomas (Eds.), *Handbook of Perception and Human Performance* (Vol. II Cognitive Processes and Performance, pp. 42-41 - 42-49). Wiley & Sons, 1986.

[30] Eamonn O'Neill and Peter Johnson. 2004. Participatory task modelling: users and developers modelling users' tasks and domains. In *Proceedings of the 3rd annual conference on Task models and diagrams (TAMODIA '04)*. ACM, New York, NY, USA, 67-74.

[31] Philippe Palanque, Rémi Bastide, Louis Dourte. Contextual Help for Free with Formal Dialogue Design. In *Proc. of HCI International* 1993.

[32] Philippe Palanque, Célia Martinie, and Camille Fayollas. 2018. Automation: Danger or Opportunity? Designing and Assessing Automation for Interactive Systems. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. ACM, New York, NY, USA, Paper C19, 4 pages.

[33] Pangoli S., Paternò F. Automatic Generation of Task-Oriented Help. *ACM Symposium on UIST 1995*, 181-187.

[34] Fabio Paterno. Task models in interactive software systems, *Handbook of Software Engineering and Knowledge Engineering*, Vol 1, 2002, Publisher: World Scientific, pp. 1-19.

[35] Fabio Paternò, Cristiano Mancini, Silvia Meniconi. ConcurTaskTrees. 1997. A Diagrammatic Notation for Specifying Task Models. In *Proc. of IFIP INTERACT 1997*, pp. 362-369.

[36] Fabio Paternò and Enrico Zini. 2004. Applying information visualization techniques to visual representations of task models. In *Proceedings of the 3rd annual conference on Task models and diagrams (TAMODIA '04)*. ACM, New York, NY, USA, 105-111.

[37] David Pinelle, Carl Gutwin, and Saul Greenberg. 2003. Task analysis for groupware usability evaluation: Modelling shared-workspace tasks with the mechanics of collaboration. *ACM Trans. Comput.-Hum. Interact.* 10, 4 (December 2003), 281-311.

[38] Martina Ragosta, Célia Martinie, Philippe Palanque, David Navarre, and Mark Alexander Sujan. 2015. Concept Maps for Integrating Modelling Techniques for the Analysis and Re-Design of Partly-Autonomous Interactive Systems. In *Proc. of the 5th International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS '15)*, ACM, 41-52.

[39] Daniel Sinnig, Maik Wurdel, Peter Forbrig, Patrice Chalin, Ferhat Khendek. Practical Extensions for Task Models. *In proc. of TAMODIA 2007*, 42-55, Springer.

[40] Gerrit C. van der Veer, Bert F. Lenting, Bas A.J. Bergevoet. 1996. GTA: Groupware task analysis — Modelling complexity, *Acta Psychologica*, Volume 91, Issue 3, pages 297-322.

[41] Marco Winckler, Philippe Palanque, and Carla M. D. S. Freitas. 2004. Tasks and scenario-based evaluation of information visualization techniques. In Proceedings of the 3rd annual conference on Task models and diagrams (TAMODIA '04). ACM, New York, NY, USA, 165-172.

[42] Palanque, P., Basnyat, S.: Task Patterns For Taking Into Account In An Efficient and Systematic Way Both Standard And Erroneous User Behaviours. In: IFIP 13.5 Working Conf. on Human Error, Safety and Systems Development (HESSD), pp. 109–130. Kluwer Academic Publisher, Dordrecht (2004)

[43] C. Fayollas, C. Martinie, P. Palanque, Y. Deleris, J. -. Fabre and D. Navarre, "An Approach for Assessing the Impact of Dependability on Usability: Application to Interactive Cockpits," *2014 Tenth European Dependable Computing Conference*, Newcastle, 2014, pp. 198-209, doi: 10.1109/EDCC.2014.17.

[44] Nicolas Broders, Célia Martinie, Philippe Palanque and Kimmo Halunen. A Generic Multimodels-Based Approach for the Usability and Security Analysis of Authentication Mechanisms. 8th International Conference on Human-Centered Software Engineering 2020, LNCS, Springer, to appear

[45] Bernhaupt R., Palanque P., Drouet D., Martinie C. (2019) Enriching Task Models with Usability and User Experience Evaluation Data. In: Bogdan C., Kuusinen K., Lárusdóttir M., Palanque P., Winckler M. (eds) Human-Centered Software Engineering. HCSE 2018.

Lecture Notes in Computer Science, vol 11262. Springer, Cham. https://doi.org/10.1007/978-3-030-05909-5_9

Steps forward towards developing preschoolers` digital literacy. An experience report

Anamaria Moldovan
Bee Kindergarten, 27,
Gr.Alexandrescu,
Cluj-Napoca,Romania
anabeekindergarten@gmail.com

Adriana-Mihaela Guran
Babes-Bolyai University
1, M.Kogalniceanu,
Cluj-Napoca, Romania
adriana@cs.ubbcluj.ro

Grigoreta-Sofia Cojocar
Babes-Bolyai University
1, M.Kogalniceanu,
Cluj-Napoca, Romania
grigo@cs.ubbcluj.ro

DOI: 10.37789/rochi.2020.1.1.2

ABSTRACT

Early childhood education (ECE) encompasses, at its most basic level, all forms of education, both formal and informal, provided to young children up to approximately 8 years of age. Some of the benefits include a diminished risk of social-emotional mental health problems and increased self-sufficiency as children mature and enter adulthood. It is during this period that children go through the most rapid phase of growth and development. Their brains develop faster than at any other point in their lives, so these years are critical. The foundations for their social skills, self-esteem, perception of the world and moral outlook are established during these years, as well as the development of cognitive skills. Lately, digital literacy and its practices are added as a need for future personal and professional development of children and developing digital competence already at kindergarten level can also help to raise awareness on safety issues and build critical thinking among them regarding content and devices that they use. This paper describes a new approach in supporting the development of digital literacy skills of preschoolers aged 5 to 6 in the context of Human-Computer Interaction (HCI) students volunteering for ICT activities in kindergarten along with building edutainment applications. There is also a report of key digital competences established by the kindergarten teachers involved, and monitored for formation during the activities.

Keywords

Education; Digital literacy; Interaction; Human; Preschoolers; Key Competences; ICT activities

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

INTRODUCTION

The Romanian educational system has made significant progress in recent decades, consolidating its institutions and improving students` learning outcomes. However, although it gives some students a chance to excel, many others do not master basic skills and almost one-fifth drop out of school before graduating from high school. Creating an educational system where all students have access to quality education and are supported to give their best will improve further performance and the process of learning, thus supporting individual well-being and growth at the national level.

Romania is currently implementing an ambitious curriculum, focused on student-guided learning and the development of key competencies. In this way, it has the opportunity to achieve a deeper transformation in terms of what is appreciated and taught in the classroom throughout the country. Strengthening the evaluation and examination system in the sense of establishing high expectations for all students but also training practices that contribute to the development of students, teachers and schools will play an essential role in achieving this transformation and creating a more equitable educational system in which all students have access to quality education [4,19].

The basis of the ability to learn throughout life is formed in the early years of childhood. Learning is a gradual process, and building strong premises in early childhood is a prerequisite for the development of skills and educational success at higher levels, being equally essential for the health and well-being of children. In this context, the goals of early education aim at a series of aspects, which frame the premises of the key competencies formed, developed and diversified, on the path of further schooling. Each European country established the level of ECE regarding the age of children and organized the educational system and the curricula according to that.

The curriculum for early education capitalizes on the curricular paradigm centered on competences and bases the educational approaches on the child and on his/ her learning activity as a process, respectively on the acquisition of behaviors that will ensure the premises for the development of key competences later. Taking into account the peculiarities of the child's development up to 6 years (the age limit for ECE in Romania), it is not appropriate to use the concept of *competence*, understood as an integrated set of knowledge, skills and attitudes [8,9,18]. Competence involves the existence of cognitive patterns and patterns of behavior, so a higher level of development than that reached by typical children in the 0-3 years and even 3-6 years. The Recommendation of the European Parliament and of the Council, 23 April 2008, regarding the European Framework Qualifications for lifelong learning [21] describes *competence* from the perspective of responsibility and autonomy as a proven ability to use personal, social and/or methodological knowledge, skills and abilities in work or study situations and for professional and personal development.

The option to use key competences as benchmarks in establishing the training profile for pre-university education was an important decision of educational policy promoted by the National Education Law no. 1/2011. Early education is established to be the heart in developing these competencies, starting from the model of structuring the development levels of eight key competence domains, in relation to the pre-university education levels. These are: communication in the native language; communication in foreign languages; mathematical skills and basic skills in science and technology; digital competence; to learn how to learn; social and civic skills; spirit of initiative and entrepreneurship; awareness and cultural expression [4,19,21].

Communication skills in foreign languages and *digital competences* are not directly covered by the early education curriculum, but it is recommended that, as far as available resources allow, they should be addressed through the carrying out activities [4].

In the next sections, we present the educational context, the strategy and the observations listed at the end of ICT activities organized with the help of volunteering HCI students for preschoolers aged 5 to 6 in terms of behaviors and attitudes towards technology (computer, laptop).

EDUTAINMENT APPLICATIONS

The role of games in learning has been studied for years ago, and it has been proven that no matter the age, mixing fun with learning is appropriate and brings valuable results. This is because the elements of games determine immersion, engagement and motivation for the learners. Moreover, when the final users of an application with an educational intended role are preschoolers, the presence of a narration or game elements become a must. Edutainment applications combine learning with playing. Designing edutainment applications is challenging because it needs to envelop the learning content and tasks in the scenario of a game [10,20]. Designing edutainment for preschoolers is even more challenging due to the constraints introduced by their development. The previous experience in designing edutainment applications [12, 14] and computer aided assessment applications brought the determination to address in a new manner the support provided for digital skills development of preschoolers.

To support the development of digital skills of preschoolers the intervention on multiple aspects is needed:

- children should be exposed to interactive applications that are appropriate for their age in terms of content, tasks and interaction actions that are required. Because there are many differences between preschoolers in the small group (3-4 years old), middle group (4-5 years old) and large group (5-6 years old) specific applications should be designed and developed for each age range;
- children should be gently introduced to the ICT world, by presenting them the fundamentals of computers, and interaction with them;
- monitoring the results of using the edutainment applications is necessary. The results should be assessed in terms of knowledge acquired by the children and development/improvement of digital skills.

In the following it is described the approach that has been used to cover all the previously mentioned aspects.

Designing developmentally appropriate edutainment applications

The development of edutainment applications for preschoolers is a challenging task from the following perspectives:

- The final users are very small children that cannot read or write and have reduced communication skills;
- Mixing education and fun needs creativity and good design and programming skills;
- Differences between the preschoolers' age groups are considerable, so preschoolers cannot be seen as a homogenous group;
- There is a lack of design guidelines that address children of 3 to 6 years; most of the recommendations are considering children between 0 and 12 years, with no differentiation;
- The use of existing user centered design methods needs adaptation due to the age of the final users;
- The assessment methods of the final products should refer to usability from the children' perspective, acceptance from kindergarten teachers and efficiency comparatively with the classic teaching method.

We have previous experience in developing edutainment applications using an adapted User Centered Design (UCD) approach [12, 13, 14, 15]. The adaptation refers to the involvement of children during the design process. In the previous work, kindergarten teachers have been proposed as *surrogate* of the children during the design alternative step, due to the abstract nature of the design sketches. Another approach, more expensive, is to merge the design alternative and prototyping step, such that children could effectively participate with feedback on the proposed solution. During the user centered design process both preschool children and educational experts (kindergarten teachers) have been involved due to the educational goal of the developed applications, while during the assessment step also the preschooler's parents were involved to create an overall image on children' attitude toward their interaction experience.

Assessing the developed education applications with the final users needs adaptation, because the methods applied with adult users cannot be applied similarly with such small children. Observation was considered as a valuable assessment method, together with smileyometers and peer tutoring. Because children are usually willing to please the adults, the use of interviews to gather their subjective opinion on the applications is not relevant. In [11] we have proposed a method to automatically identify children' emotion during the interaction. The main focus was on identifying the presence of negative emotions and the interaction context where they have occurred, to guide our future redesign decisions.

Introducing ICT concepts to preschoolers

After having the experience of designing and developing edutainment applications for preschoolers the conclusion that it would be helpful to prepare children for their interaction tasks was drawn. Although the current generation of preschoolers are digital natives, they have reduced experience with personal computers and/or laptops. The idea that children should be introduced to these devices, to know basic things about their components and roles along with using the applications became a must do. In this perspective, the curriculum for an optional ICT class called "A computer to learn!"[2, 3, 5, 6, 16, 17] was conceived with the help of kindergarten teachers involved. The curriculum approaches the following subjects: understanding of devices we use, the components of a personal computer, input and output devices, tasks from children` usual activities that can be automated using the computers (drawing, typing letters and digits, and coloring) [3, 20, 22]. Similar to the design of edutainment applications tasks, all the activities were conceived as games. Support in teaching the optional course has been received from the third year students from the Faculty of Mathematics and Computer Science attending the Human-Computer Interaction optional course. All of the students were involved at the same time in the design and development of edutainment applications. Teams of 3 to 5 students have interacted with groups of 20-25 preschoolers in 40 minutes sessions/week for six weeks. During the ICT optional classes children have been introduced to theoretical information on the basic concepts presented before (see Figure 1.) followed by a practical part where they have had the chance to practice fundamental interaction tasks (using the mouse and the keyboard to perform various tasks as Figure 2. shows).

Each team of students organized an activity using intuitive materials as drawings, pictures and their own devices. They also organized games such as a role-play with four children involved-one was the child, one the mouse, one the hard, one the file on the desktop; When the child pressed the left side of the mouse giving an order, the hard processed and the file opened. Another played game organized children in four groups and four children at the same time were supposed to move according to the arrow shown by one of the students that held the activity. The movements stopped when the first child in the row reached a piece of paper where it was written ENTER.

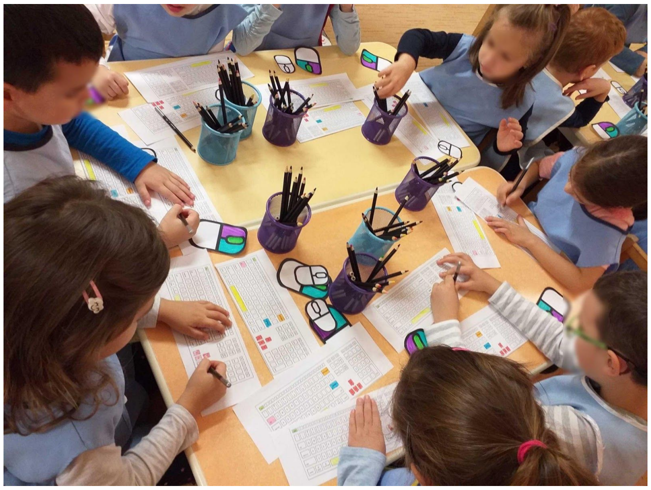


Figure 1. Activity example for introducing the keyboard and the mouse

All the actions imitated during games were then practised on real devices: computer and laptops. Many pictures were taken for parents to see a part of the activities as they all agreed to their children participation and the use of them for academic purposes as articles or presentations.

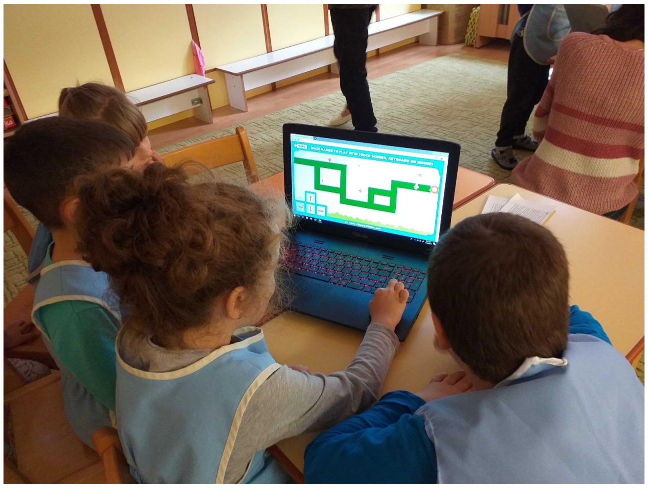


Figure 2. Practical activity

Following the nurturing of digital skills

Focusing the educational process on the child, a fully accepted principle of contemporary education, implies essentially the permanent concern of teachers for the knowledge of the child as individuality and adaptation of educational programs to the individual profile. From this perspective, the consideration of the child as a whole, as a structured set of defining features, needs, inclinations, potential in a close determination and mutual relations is an essential condition for the formation, through specific means of education, of integral and harmonious personalities [4, 5, 18].

This implies, first of all, the recognition of the child as an individuality, a personality in formation, whose areas of manifestation - physical, spiritual, emotional, cognitive, social -, influence each other and develop simultaneously, each of these being equally important and having to be the object of early education. Consequently, the edutainment applications offered during the education period from 1 to 6 years must identify and use curriculum sequences that address not only the cognitive area of the personality, but also the affective, social and motor ones, so that the development in a certain side can support and stimulate the evolution of the other sides and the personality as a whole.

The unique character of the child's personality is also given by the specifics of the individual needs of knowledge and training of the child which are considered more and more often, in contemporary pedagogy, the starting point of the educational intervention [4, 5, 18].

As ECE gained importance and all the recommendations for future socio-economic development of EU countries began from this point, the educational policies changed and adapted their curriculas in terms of key competencies to be followed for formation during the entire school period and further. Preschool stage, kindergarten time is now considered the heart of the whole educational system so the curriculum has been adapted to cover the newly promoted tendencies.

Due to peculiarities of the child's development up to 6 years, it is not appropriate to use the concept of competence, understood as an integrated set of knowledge, skills and attitudes, but behaviors and attitudes separately can be established and purchased in achieving.

Also, *digital competences* are not directly covered by the early education curriculum in Romania, but it is recommended that, as far as available resources allow, they should be addressed through the carrying out activities. The context of designing appropriate edutainment applications and of introducing ICT concepts to preschoolers proved to be a resourceful one in observing the nurturing of preschoolers` digital skills.

Behaviour covers knowledge and skills used by the individual in certain specific situations. If it is noticeable, it can be evaluated through the quality of the performing actions and the quality of the results [5, 8, 9].

As a mandatory task for the approval of the ICT class, there were formulated actions in terms of behaviours, complying with preschoolers` needs and features [4, 5, 18] along with content and goals, all these by the kindergarten teachers involved during activities. Behaviours stated into the curricula were considered a prediction of what children should be able to perform at the end of the activities and gave the teacher the possibility to observe and list their occurrence. Thus, twenty kindergarten children were monitored during the six weeks of ICT activities, through observation. Interviews have been organized and the key-word in the frequency of occurring behaviors was considered: *constantly* - the child does all the actions *alone* or the child does the actions with help-*only helped*. The choice of the word was not random but correlated to the specific of the behaviors which, in patterns, develop into competence [5, 8, 9].

FINDINGS

Although there is not yet the case of extracting data from an experiment and much work stands ahead, it can surely be stated that the goals established for the ICT class [2] were achieved. Behaviors mentioned below, and their observed frequency could be a starting point for further actions and improvements, as all actors involved in the process of education, teachers, children, parents, community, proved to be more than supportive.

Frequency

Observed/monitored behaviors	**Constantly-alone/Total number of children**	**Constantly-only helped/Total number of children**
Recognizes and differentiate the components of a computer: screen, keyboard, mouse	20/20	0/20
Interacts: turn on/turn off the computer, click execution, double-click, right click, drag & drop, mouse selection, usage of ENTER, SPACEBAR, ESC buttons, open an application, close an application	8/20 but not all the behaviors at the same time	12/20
Recognizes elements of interaction in WIMP interfaces	4/20	16/20
Recognizes interaction graphics and their meaning (menu, button, radio button, checkbox, text field)	5/20	15/20
Uses basic commands in Microsoft Paint-choosing colours, cutting and copying, choosing graphics	4/20	16/20
Responds to a command, performing the interaction	16/20	4/20
Expresses the desire to perform a certain interaction	20/20	0/20
Demonstrates perseverance in performing a task	13/20	7/20
Verbalizes the performed interaction	0/20	20/20

Table 1: Listed behaviours and their occurrence during ICT activities

During the activities they performed in the kindergarten, whether it was an ICT one or the edutainment application assessment, students have remarked and underlined the joy, enthusiasm and willingness of children to interact with them and their products, to explore and to replay the games, challenging and motivating them.

On the other side, children gained new playing experience and a proper attitude towards technology translated into: ”computers are built and organized by people and we are the ones who learn WHAT to do with them from the ones who know HOW!”. They even integrated their new abilities into their role-play games: they built laptops and computers and added a mouse, performed an imaginary click and opened an imaginary file. Step-by-step they internalized the actions and the vocabulary.

CONCLUSIONS AND FURTHER WORK

This paper is a report of a challenging initiative to combine the process of developing edutainment applications with organizing ICT activities for the same users: preschool children aged 5 to 6, as a step forward towards their future literacy. In the future we intend to extend this approach to identify an appropriate method to rigorously evaluate the effectiveness of using the edutainment applications in terms of learning outcomes in comparison to the classical teaching approach. There is also the intention to focus more on ICT activities organized for preschoolers in order to provide and support the development of a proper attitude towards technology and digital skills. And, an important aspect, we intend to collaborate with other experts in education, teachers, researchers, to enhance the methodological aspects and to validate the results.

ACKNOWLEDGEMENTS

We would like to thank all the children and their parents, kindergarten teachers and students who participated in this proposal. We would also like to thank for the given support to the kindergarten management team who fully supported this approach in nurturing the digital skills of preschoolers.

REFERENCES

1. Bekker T., Markopoulos P. *Interaction design and children*. In:Interacting with Computers 15, 2003

2. Bloom, Benjamin, *Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook I: Cognitive domain..* New York: David McKay Company,1956

3. Chaudron, S., Di Gioia, R., Gemo, M., *Young Children (0-8) and Digital Technology. A Qualitative study across Europe,* European Commission, 2018

4. *Curriculum pentru Educaţie timpurie*, MEN, Bucureşti, 2019, https://www.edu.ro/sites/default/files/Curriculum%20ET_2019_aug.pdf

5. Bogaert, C., Delmarle, S&Preda, V., *Formarea competenţelor în grădiniţă. O altă perspectivă asupra timpului şcolar.* Ed. Aramis, Bucureşti, 2013

6.Creer, A. *Introducing Everyday'Digital Literacy Practices'into the Classroom: an Analysis of Multi-layered Media, Modes and their Affordances*. Journal of New Approaches in Educational Research, 7(2), 131-139. doi: 10.7821/naer.2018.7.265, 2018

7.Crescenzi, L., Gran, M., *An Analysis of the Interaction Design of the Best Educational Apps for Children Aged Zero to Eight*. Comunicar, 46, 77-85.

8. Dulamă,E., *Fundamente despre competenţe, Teorie şi aplicaţii,* Presa Universitară Clujeană, Cluj-Napoca, 2010

9. Dulamă, E., *Despre competenţe, Teorie şi practică*, Presa Universitară Clujeană, Cluj-Napoca

10. Grasset, Raphaël, Dünser, Andreas, Billinghurst, Mark, *Edutainment with a mixed reality book: a visually augmented illustrative childrens' book*, ACE'08 Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology, Pages 292-295

11. Guran, A. M. Cojocar, G. S., and Dioşan L. S. 2020. *A Step Towards Preschoolers' Satisfaction Assessment Support by Facial Expression Emotions Identification*. In the 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems. to be published.

12. Guran, A. M. Cojocar, G. S. , and Moldovan, A. 2018. *Initiative to Support Basic Digital Skills Development of Romanian Preschool Children.* In 15th International Conference on Human Computer Interaction, RoCHI 2018, Cluj-Napoca, Romania, September 3-4, 2018, Adrian Sabou and Philippe A. Palanque (Eds.). Matrix Rom, 147–154.

13. Guran, A.M. Cojocar, G. S. and Moldovan A. M. 2019. *Applying UCD for Designing Learning Experiences for Romanian Preschoolers. A Case Study. In Human-Computer Interaction* - INTERACT 2019 - 17th IFIP TC 13 International Conference, Paphos, Cyprus, September 2-6, 2019, Proceedings, Part IV (Lecture Notes in Computer Science, Vol. 11749), David Lamas, Fernando Loizides, Lennart E. Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris (Eds.). Springer, 589–594. https://doi.org/10.1007/978-3-030-29390-1_43

14. Guran, A.M. Cojocar, G. S. and Moldovan A. M. 2019. *Co-Design of Edutainment Applications with Preschoolers. Is it feasible?*. In 16th International Conference on Human-Computer Interaction, RoCHI 2019, Bucharest, Romania, October 17-18, 2019, Alin Moldoveanu and Alan J. Dix (Eds.). Matrix Rom, 92–97.

15. Guran, A.M. Cojocar, G. S. and Moldovan A. M. 2020. *A User Centered Approach in Designing Computer Aided Assessment Applications for Preschoolers.* In Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2020, Prague, Czech Republic, May 5-6, 2020, Raian Ali, Hermann Kaindl, and Leszek A. Maciaszek (Eds.). SCITEPRESS, 506–513. https://doi.org/10.5220/00095655050605

16. Hourcade J. P. Interaction Design and Children Found. *Trends Hum.-Comput. Interact*. 1, 4 (April 2008), 277-392.

17. Markopoulos, P., Read, J., MacFarlane, S., *Evaluating Children's Interactive Products. Principles and Practices for Interaction Designers*, 2008

18. Piaget, Jean, *Şase studii de Psihologie*, 3 Publishing House, 2017

19. Raport OECD - UNICEF: *Evaluările şi examinările în sistemul de educaţie din România*, https://www.edu.ro/raport-oecd-unicef-evalu%C4%83rile-%C8%99i-examin%C4%83rile-%C3%AEn-sistemul-de-educa%C8%9Bie-din-rom%C3%A2nia, Iunie 2020

20. Rapeepisarn, Kowit, Wong, Kok Wai, Fung Chun Che, Depickere, Arnold, *Similarities and differences between "learn through play" and "edutainment,* *https://pdfs.semanticscholar.org/f92a/aa7f6d54d2ca98b6f64122f89d0beb6fd1ee.*, Iunie *2019*

21. Recommendation of the European Parliament and of the Council of 23 April 2008 on the establishment of the European Qualifications Framework for lifelong learning (Text with EEA relevance)*OJ C 111, 6.5.2008, p. 1–7*

22. Venn-Wycherley, Megan, Kharrufa, Ahmed, *HOPE for Computing Education: Towards the Infrastructuring of Support for University-School Partnerships,* CHI '19 Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019

The Lib2Life Platform – Processing, Indexing and Semantic Search for Old Romanian Documents

Irina Mitocaru, Gabriel Guțu-Robu, Melania Nițu, Mihai Dascălu, Ștefan Trăușan-Matu

University Politehnica of Bucharest

Splaiul Independentei 313, 060042, Bucharest, Romania

{irina.mitocaru, suzana_melania.nitu}@stud.acs.pub.ro
{gabriel.gutu, mihai.dascalu, stefan.trausan}@upb.ro

Silvia Tomescu

Carol I Central University Library Bucharest

Boteanu 1, 010027, Bucharest, Romania

silvia.tomescu@bcub.ro

Gabriela Florescu

National Institute for Research and Development
in Informatics ICI Bucharest

Maresal Averescu 8-10, 011455, Bucharest, Romania

gabriela.florescu@ici.ro

DOI:10.37789/rochi.2020.1.1.3

ABSTRACT

Preserving the cultural heritage of a nation throughout generations is essential in a continuously developing society. This paper introduces the Lib2Life platform powered by advanced Natural Language Processing techniques, focusing on the processing, indexing and semantic search of old documents from the Central University Libraries in Romania. Our platform enables the upload and text pre-processing of scanned documents by librarians, who can afterwards manually correct the extracted content and corresponding metadata. In addition, Lib2Life ensures the exploration of the collection of books using a semantic search engine to retrieve documents fitted to the users' interests. The platform was evaluated using an usability questionnaire which pinpointed out that Lib2Life is a modern and user-friendly smart search engine for old documents written in the Romanian language. Improvements in terms of server response time and functionality were suggested. The platform proved to be intuitive and easy to use, having the potential to become an analytical system incorporating a rich and diverse collection of books.

Author Keywords

Library software; automated text extraction; Natural Language Processing.

ACM Classification Keywords

H.3.7: Information Systems: Information Storage and Retrieval: Digital Libraries.

H.5.1: Information Systems: Information Interfaces and Presentation: Multimedia Information Systems

General Terms

Text analysis; Software usability.

INTRODUCTION

The evolution of technology nowadays allows people to use electronic devices to access digitalized documents. Digitalization becomes mandatory due to the increased ubiquity of electronic devices, and their ease of use compared to traditional reading, as well as document research methods.

The Central University Libraries from Romania are dispersed throughout the country and host large numbers of old documents that are no longer copyrighted. These documents include books, manuscripts, or newspapers, and they serve to better understand life during those times in terms of politics, science, or education. Digital documents safeguard the initial manuscripts from being naturally deteriorated and provide an unlimited lifespan within a virtual environment. Moreover, access to the original documents can be limited to preserve degradation through manual handling. In contrast, digital documents can be stored in a convenient format that does not take too much storage space, such as the PDF. PDF documents have a structured format and contain also metadata, which is used either to properly display the document to the user (such as font types, font colors, or the physical coordinates of the words), or to store other annotations. Digitalization also eliminates distance and time limitations by allowing concurrent access to documents for individuals, regardless of their physical location or their time availability.

This paper describes the Lib2Life platform, which integrates Natural Language Processing (NLP) techniques and allows Romanian Central University libraries to store digitalized documents and provide access to resources to the public. The platform relies on a processing pipeline designed to properly support the extraction of texts from scanned documents. The paper continues with the description of related systems and text extraction applications. Then follows a presentation of our platform, its evaluation, conclusions and future work.

STATE OF THE ART

A well-known web application used for accessing online books is Google Books (http://books.google.com), which includes a large collection of books retrieved from different data sources. Google Books indexes data on their servers and provides access to portions of texts from the books. Links to buy or to allow further reading the entire document are included. Google Books is equipped with a smart search engine that uses NLP techniques to retrieve the most relevant books and corresponding passages, given the user query.

Google also provides a facility named Talk to Books, which compares the user query with every sentence in over 100,000 books to find responses that would most likely be related with that text or that could be an answer to the user's query. The suitable documents are then retrieved and shown in bold to the user, together with portions of text to provide a better contextualization. This approach came from the idea of mimicking a real conversation by using billions of lines of dialogue to teach an AI how human conversations flow. Their model can predict how likely a statement would follow another as a response based on a collection of possible responses [9; 14]. Similarly, the user can search for books in the Lib2Life platform using a free input text field, which is then semantically compared with portions of texts from the indexed books. Moreover, the user has the ability to find similar books using semantic similarity. The process is detailed further on in the Method section.

At national level, existing solutions to access digitalized documents from Central University Libraries (CULs) are outdated and provide rather limited functionalities (e.g., no search within the actual documents). For example, the Carol I Central University Library relies on Vubis [1] to save various metadata and Restitutio (http://restitutio.bcub.ro/), which is based on DSpace [15]. CUL Cluj has created their online catalog (http://aleph.bcucluj.ro:8991) designed to help people find the physical location of a book easier when that desired book is available in the library. Users have to provide search metadata, such as the book's identifier, the name of the author, the book's title, the publishing house, or the publishing year. Multiple databases are available, like book catalogues or bibliographies. Results can be further restricted by using specific filters, such as the language, time constraints (specific publication period), or the book's publication domain. BCU Cluj also has a digital library (http://dspace.bcucluj.ro) based on DSpace that provides online access to books contained in the physical library. The platform is accessible through a user-friendly and more intuitive interface than the interface provided by the Vubis catalog system. Each book can be read online using a PDF reader incorporated in the browser, or it can be downloaded on user's computer. The intuitiveness behind the incorporated filters and the multitude of options also served as an inspiration for developing the Lib2life platform, which incorporates a similar approach of filtering criteria for the semantic search engine.

Recommender systems are frequently applied in domains like online shopping and entertainment to predict user preferences. However, the same approach can be used to recommend books by relying on user profiles [13]. Their proposed recommended system takes into account multiple aspects for matching a book with a user. The first aspect consists of matching other users' interests, while the second refers to considering the temporal dimension. Specifically, the temporal dimension relies on the fact that user's preferences change over time.

The previously presented systems helped us into shaping the Lib2Life platform based on the requirements of librarians. Lib2Life provides three major functionalities. First, the system's focus is to centralize documents from multiple sources, more specifically from Romanian Central University Libraries, offering a single point of access for their entire data. Second, Lib2Life incorporates a multitude of NLP approaches, like a semantic search engine. Third, the platform also relies on an ontology for properly categorizing documents.

METHOD

Corpora

The Central University Libraries in Romania built up a collection of about 2,000 scanned documents written in Romanian. This dataset consists mostly of books dated in the 19th or 20th century scanned using high resolution scanners. However, part of the collection was not in a proper format for applying Optical Character Recognition (OCR) due to human errors or scanner issues. In addition, the OCR process encountered problems due to the limited resolution of the documents in some cases, or their degraded physical format.

The OCR process relies on the Tesseract API [16] applied on scanned documents before uploading them to the Lib2Life platform. The API was adapted to allow a good precision in detecting characters and to compress the file into a rather small size PDF document in the end. This requirement also resulted after an iterative process of understanding limitations and improving the general workflow. Large documents (i.e., tens of megabytes) were time consuming when uploaded to the Lib2Life platform, when processing them, and also when accessing them via the integrated PDF viewer in the web browser. Currently, the size of an OCR-ized document is about 10-15 MB.

The target of Lib2Life is to share the cultural heritage of millions of processed historical pages existent in CULs. Nevertheless, the Carol I Central University currently hosts about 2.4 millions of volumes (http://www.bcub.ro/colectii). However, part of the collection contains documents with publishing rights which cannot be used in our platform.

Architecture

The Lib2Life platform contains a web portal that allows librarians and users to interact with the system. The architecture of the platform is presented in Figure 1. The backend of the system integrates several digital services, for example: 1) document categorization; 2) semantic search; 3) semantic recommendations of similar documents. Assigning a category to the document is performed after uploading it and setting its corresponding metadata. The service is based on the Lib2Life ontology [7], which incorporates several domains and the relations established between them. The second service includes the facility to search the indexed documents using filtering criteria or keywords. Semantic algorithms are used to find documents matching the user's keywords. The third service consists of semantic recommendations – similar documents with the accessed one are provided. Both document search and semantic recommendations rely on Elasticsearch indexing [6]. Elasticsearch (https://www.elastic.co) is a non-relational database that stores and indexes the documents' metadata and their content using the JSON format. Elasticsearch provides fast and easy to use queries, filters, and aggregations mechanisms, which were incorporated in the search functionalities provided by the Lib2Life platform.

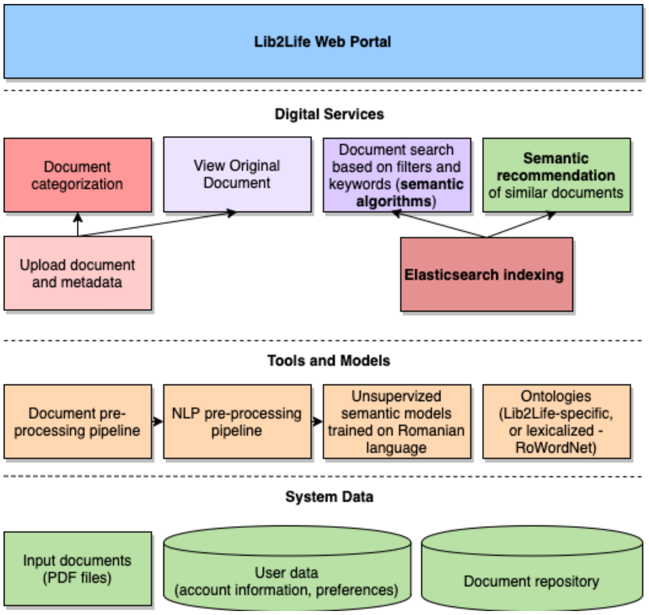

Lib2Life Web Portal
Digital Services
Document categorization
View Original Document
Document search based on filters and keywords (semantic algorithms)
Semantic recommendation of similar documents
Upload document and metadata
Elasticsearch indexing
Tools and Models
Document pre-processing pipeline
NLP pre-processing pipeline
Unsupervized semantic models trained on Romanian language
Ontologies (Lib2Life-specific, or lexicalized - RoWordNet)
System Data
Input documents (PDF files)
User data (account information, preferences)
Document repository

Figure 1. Lib2Life platform architecture.

Tools and models from the ReaderBench framework (http://readerbench.com) [8] are used to provide the previous services. The document pre-processing pipeline refers to the extraction of metadata from the OCR-ized document. The NLP pre-processing pipeline consists of several steps, such as tokenization, part of speech tagging, and lemmatization. The considered unsupervised semantic models include Latent Semantic Analysis [4], Latent Dirichlet Allocation [2], and word2vec [11]. These models are trained on language-specific corpora of documents.

The last layer from the Lib2Life architecture considers data modeling, namely: OCR-ized PDF documents, personal users' data used to interact with the Lib2Life web portal, as well as information related to documents indexed in Elasticsearch.

Document Pre-Processing Workflow

Prior to indexing documents in Elasticsearch, text preprocessing steps are applied. The input data consists of old scanned books on which Optical Character Recognition (OCR) is applied. The OCR process brought challenges in order to allow proper extraction of texts. These included different font types and sizes identified in the same section or line of text, different styles for headers and footers in the same document, disruption of paragraphs, improper page breaks, loss of content structure, or misinterpretation of certain characters and hyphenated words. Currently existing systems are not designed to work with OCR-ized PDFs [12], raising challenges while trying to properly restructure the recognized text. The identified issues imposed the necessity of a workflow that can identify and correlate section titles with their content, recognize paragraphs boundaries, merge hyphenated words and accurately identify and extract images or tables. The Lib2Life document processing workflow (see Figure 2) is designed to index documents into Elasticsearch and facilitate the search for relevant resources based on keywords.

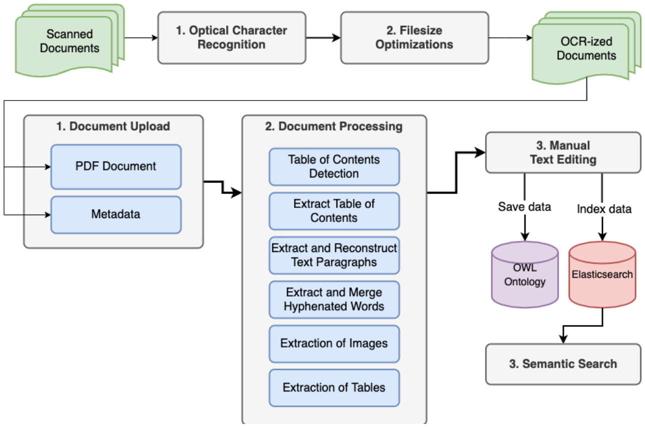
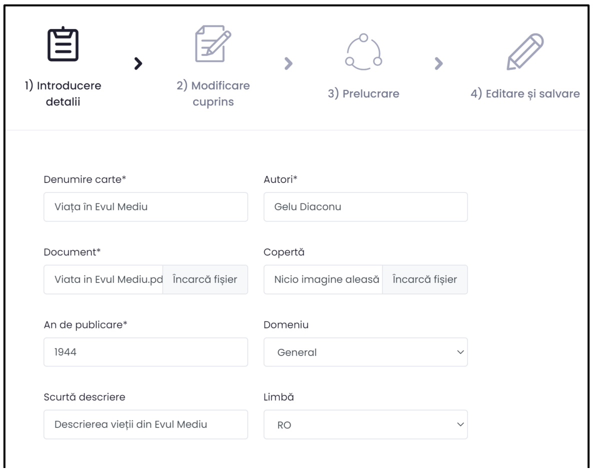


Figure 2. Lib2Life document processing workflow.

Documents are parsed line by line, identifying relevant sections and metadata within the document such as section titles, section headings, paragraphs, images, tables, and the table of contents. Paragraph boundaries are reconstructed, and hyphenated words are merged. The document title, the author, and the publishing year are extracted from the first page (if they are available) and are passed to the processing phase. The librarian has the facility to manually introduce a document's metadata when uploading the file – see Figure 3.

After the initial upload, three steps are performed: 1) detection and extraction of the table of contents and corresponding text; 2) information extraction using NLP techniques and heuristics; and 3) manual text editing.



Denumire carte*
Viața în Evul Mediu

Autori*
Gelu Diaconu

Document*
Viata in Evul Mediu.pd Încarcă fișier

Copertă
Nicio imagine aleasă Încarcă fișier

An de publicare*
1944

Domeniu
General

Scurtă descriere
Descrierea vieții din Evul Mediu

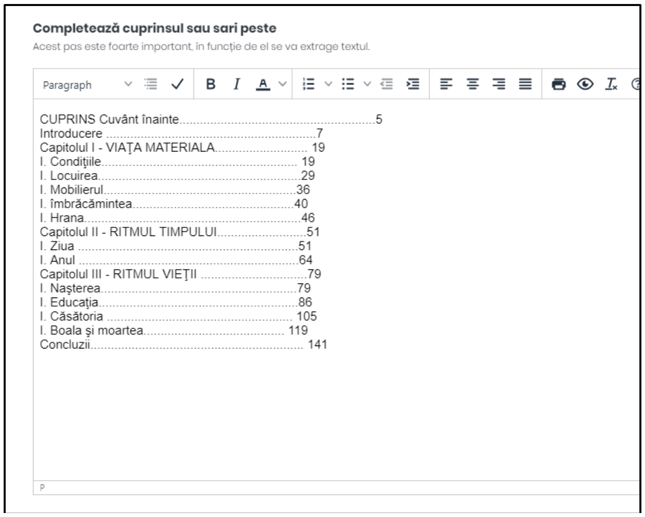
Limbă
RO

Figure 3. Document uploading and corresponding metadata.

Table of Contents Extraction

Two approaches were considered for extracting the Table of Contents (TOC): 1) correlating section titles with their content; 2) finding the predominant font type. The TOC extraction is performed by identifying the first page of the TOC within the first or last pages of a document. Specific words from the Romanian dictionary, such as „cuprins”, „tabela”, or „tabla de materii”, are looked for. The OCR-ized text may contain errors like white spaces or symbols or may be split on several lines, which imposed additional validations using regular expressions. Mapping the section title with its corresponding page number was done in accordance with lines ending with digits. Additionally, empirical values were set for the number of lines ending with digits (more than 3), the max number of pages for TOC (10 pages). TOC entries are then parsed using regular expressions to extract the section title and the associated page range. Figure 4 shows an extracted TOC.

For documents that lack the presence of TOC, the second approach is used in order to identify sections and paragraphs based on the most common font available in the document.



Completează cuprinsul sau sari peste

Acest pas este foarte important, în funcție de el se va extrage textul.

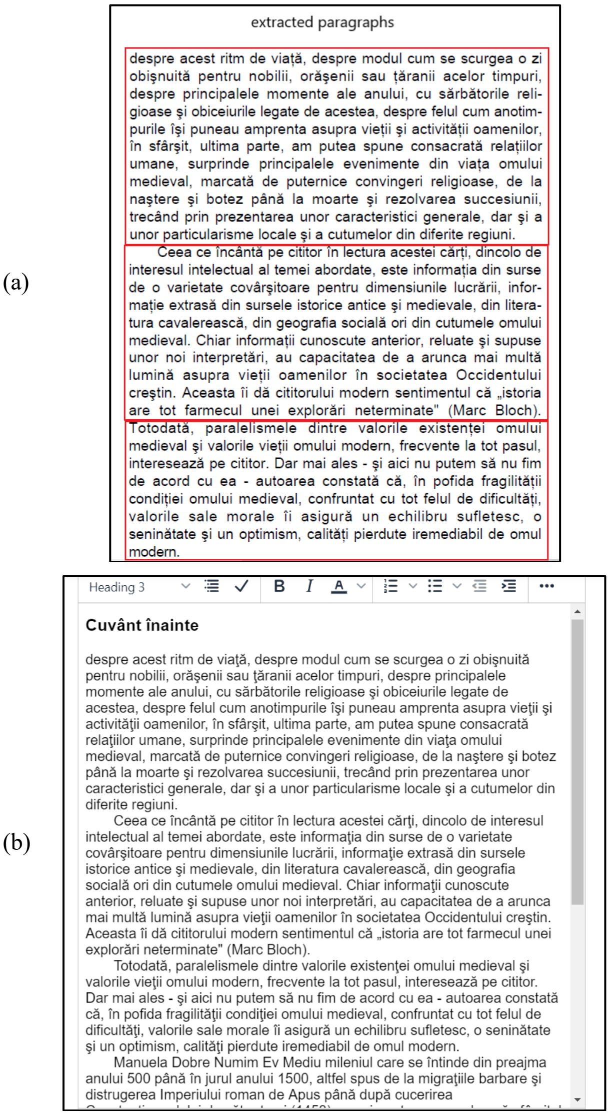
CUPRINS Cuvânt înainte...5
Introducere 7
Capitolul I - VIAŢA MATERIALA.......................... 19
I. Condiţiile.. 19
I. Locuirea..29
I. Mobilierul..36
I. îmbrăcămintea...40
I. Hrana..46
Capitolul II - RITMUL TIMPULUI.........................51
I. Ziua 51
I. Anul 64
Capitolul III - RITMUL VIEŢII 79
I. Naşterea...79
I. Educaţia...86
I. Căsătoria 105
I. Boala şi moartea....................................... 119
Concluzii.. 141

Figure 4. Table of Contents Extraction

Font name, font size, and text positions are stored in a list that is later on used to identify the type of the text (section title or body content), by comparing each line of text with the predominant font existing within the page. The two models were combined into a robust text extraction algorithm [12], that can easily adapt to most of the PDF document formats. The extracted text is then displayed in a rich text editor, enabling librarians to improve the extracted content by manually modifying the text.

Extracting and Reconstructing Paragraphs

At least one of the following conditions must be satisfied to detect a paragraph: 1) the previous line marks an ending sentence and the current line begins with an uppercase character; 2) the current line starts with a hyphen, depicting a conversation. A comparison of the original text versus the extracted text is shown in Figure 5.



extracted paragraphs

despre acest ritm de viață, despre modul cum se scurgea o zi obişnuită pentru nobilii, orăşenii sau țăranii acelor timpuri, despre principalele momente ale anului, cu sărbătorile religioase şi obiceiurile legate de acestea, despre felul cum anotimpurile îşi puneau amprenta asupra vieții şi activității oamenilor, în sfârşit, ultima parte, am putea spune consacrată relațiilor umane, surprinde principalele evenimente din viața omului medieval, marcată de puternice convingeri religioase, de la naştere şi botez până la moarte şi rezolvarea succesiunii, trecând prin prezentarea unor caracteristici generale, dar şi a unor particularisme locale şi a cutumelor din diferite regiuni.

Ceea ce încântă pe cititor în lectura acestei cărți, dincolo de interesul intelectual al temei abordate, este informația din surse de o varietate covârşitoare pentru dimensiunile lucrării, informație extrasă din sursele istorice antice şi medievale, din literatura cavalerească, din geografia socială ori din cutumele omului medieval. Chiar informații cunoscute anterior, reluate şi supuse unor noi interpretări, au capacitatea de a arunca mai multă lumină asupra vieții oamenilor în societatea Occidentului creştin. Aceasta îi dă cititorului modern sentimentul că „istoria are tot farmecul unei explorări neterminate" (Marc Bloch).

Totodată, paralelismele dintre valorile existenței omului medieval şi valorile vieții omului modern, frecvente la tot pasul, interesează pe cititor. Dar mai ales - şi aici nu putem să nu fim de acord cu ea - autoarea constată că, în pofida fragilității condiției omului medieval, confruntat cu tot felul de dificultăți, valorile sale morale îi asigură un echilibru sufletesc, o seninătate şi un optimism, calități pierdute iremediabil de omul modern.

Cuvânt înainte

despre acest ritm de viaţă, despre modul cum se scurgea o zi obişnuită pentru nobilii, orăşenii sau ţăranii acelor timpuri, despre principalele momente ale anului, cu sărbătorile religioase şi obiceiurile legate de acestea, despre felul cum anotimpurile îşi puneau amprenta asupra vieţii şi activităţii oamenilor, în sfârşit, ultima parte, am putea spune consacrată relaţiilor umane, surprinde principalele evenimente din viaţa omului medieval, marcată de puternice convingeri religioase, de la naştere şi botez până la moarte şi rezolvarea succesiunii, trecând prin prezentarea unor caracteristici generale, dar şi a unor particularisme locale şi a cutumelor din diferite regiuni.

Ceea ce încântă pe cititor în lectura acestei cărţi, dincolo de interesul intelectual al temei abordate, este informaţia din surse de o varietate covârşitoare pentru dimensiunile lucrării, informaţie extrasă din sursele istorice antice şi medievale, din literatura cavalerească, din geografia socială ori din cutumele omului medieval. Chiar informaţii cunoscute anterior, reluate şi supuse unor noi interpretări, au capacitatea de a arunca mai multă lumină asupra vieţii oamenilor în societatea Occidentului creştin. Aceasta îi dă cititorului modern sentimentul că „istoria are tot farmecul unei explorări neterminate" (Marc Bloch).

Totodată, paralelismele dintre valorile existenţei omului medieval şi valorile vieţii omului modern, frecvente la tot pasul, interesează pe cititor. Dar mai ales - şi aici nu putem să nu fim de acord cu ea - autoarea constată că, în pofida fragilităţii condiţiei omului medieval, confruntat cu tot felul de dificultăţi, valorile sale morale îi asigură un echilibru sufletesc, o seninătate şi un optimism, calităţi pierdute iremediabil de omul modern.

Manuela Dobre Numim Ev Mediu mileniul care se întinde din preajma anului 500 până în jurul anului 1500, altfel spus de la migraţiile barbare şi distrugerea Imperiului roman de Apus până după cucerirea

Figure 5. Part of the a) original text contained within the document versus b) text extracted by the algorithm.

Image Extraction

The image extraction task is based on two approaches: 1) parsing image identifiers, and 2) searching for unusual shapes within a page and identifying the number of contained colors. For example, if a page contains only text in the middle top area and in the middle bottom of a specific area, with an irregularly shaped rectangle designed on white, gray, and black colors, then most probably the rectangle is an image, even if no identifier is found. Image and text extraction tasks were joined into one document iteration to reduce the time complexity; thus, image extraction is automatically applied. When parsing the text, three heuristics were applied. First, if a designated word for figure descriptions (e.g., "Figura" or "Fig." in Romanian) is encountered, the location of the figure is saved. Second, we conducted an experiment on the number of characters existing in a page, which revealed the images were found on pages with less than 200 characters. Thus, the second heuristic compares the similarity of pixels and is applied on pages with no characters: if the pixels on the page are identical, the page is considered to be blank and it is thus skipped. Third, one of the following conditions must be satisfied to mark an entire page as an image: 1) the number of characters is zero and the pixels are different; 2) the character count is less than 200 and the text contains the figure identifier caption. The extracted images are converted to the base 64 format and the text caption is inserted into an HTML tag at the end of each section or at the end of the book, depending on the identified document structure.

Extraction of Tables

The table extraction task raised several challenges due to improper state of the OCR-ized PDF documents. An analysis performed on the collection of documents showed that most tables contained un-aligned lines, missing data, or an irregular structure. Some documents contained hand-written tables, on which the existing APIs are not able to accurately detect table boundaries, or the content itself. The current table extraction algorithm is Nurminen Detection Algorithm from Tabula API (https://github.com/tabulapdf), which showed an average accuracy of 40%. If a TOC is detected, table extraction is applied on each page, storing the coordinates of all tables and mapping the table with the corresponding section. If no TOC found, the details of the detected tables are stored, mapped with page numbers, and appended at the end of the section or of the book, depending on the identified document structure.

In addition, text cleaning steps are applied after extracting text from the document. These steps include removal of empty lines, removal of leading trailing spaces and other delimiter characters, concatenation of hyphenated words, appending white spaces for lines ending without whitespace, skipping pages with less than 400 characters (or 60 words, as in (Foundant, n.d.). Figure 6 shows an example of a page skipped because it contained too few characters.

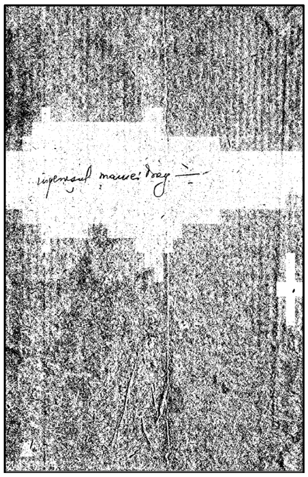
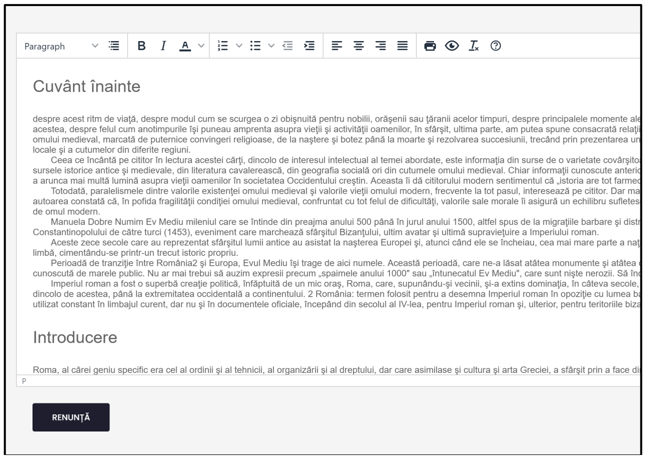


Figure 6. Page skipped because it contains too few characters.

Manual Content Editing

The refined text is sent to the user interface in an editable rich text area using TinyMCE (https://www.tiny.cloud), which enables the user to modify the extracted text before saving the file in Elasticsearch– see Figure 7. The processed text is then converted to JSON and sent to Elasticsearch for indexing. The indexed documents are used in advanced features, such as the keywords-based and semantic search.



Cuvânt înainte

despre acest ritm de viaţă, despre modul cum se scurgea o zi obişnuită pentru nobilii, orăşenii sau ţăranii acelor timpuri, despre principalele momente ale
acestea, despre felul cum anotimpurile îşi puneau amprenta asupra vieţii şi activităţii oamenilor, în sfârşit, ultima parte, am putea spune consacrată relaţii
omului medieval, marcată de puternice convingeri religioase, de la naştere şi botez până la moarte şi rezolvarea succesiunii, trecând prin prezentarea un
locale şi a cutumelor din diferite regiuni.
Ceea ce încântă pe cititor în lectura acestei cărţi, dincolo de interesul intelectual al temei abordate, este informaţia din surse de o varietate covârşito
sursele istorice antice şi medievale, din literatura cavalerească, din geografia socială ori din cutumele omului medieval. Chiar informaţii cunoscute anterio
a arunca mai multă lumină asupra vieţii oamenilor în societatea Occidentului creştin. Aceasta îi dă cititorului modern sentimentul că „istoria are tot farmec
Totodată, paralelismele dintre valorile existenţei omului medieval şi valorile vieţii omului modern, frecvente la tot pasul, interesează pe cititor. Dar ma
autoarea constată că, în pofida fragilităţii condiţiei omului medieval, confruntat cu tot felul de dificultăţi, valorile sale morale îi asigură un echilibru sufletes
de omul modern.
Manuela Dobre Numim Ev Mediu mileniul care se întinde din preajma anului 500 până în jurul anului 1500, altfel spus de la migraţiile barbare şi distr
Constantinopolului de către turci (1453), eveniment care marchează sfârşitul Bizanţului, ultim avatar şi ultimă supravieţuire a Imperiului roman.
Aceste zece secole care au reprezentat sfârşitul lumii antice au asistat la naşterea Europei şi, atunci când ele se încheiau, cea mai mare parte a naţ
limbă, cimentându-se printr-un trecut istoric propriu.
Perioadă de tranziţie între România2 şi Europa, Evul Mediu îşi trage de aici numele. Această perioadă, care ne-a lăsat atâtea monumente şi atâtea o
cunoscută de marele public. Nu ar mai trebui să auzim expresii precum „spaimele anului 1000" sau „întunecatul Ev Mediu", care sunt nişte nerozii. Să înc
Imperiul roman a fost o superbă creaţie politică, înfăptuită de un mic oraş, Roma, care, supunându-şi vecinii, şi-a extins dominaţia, în câteva secole,
dincolo de acestea, până la extremitatea occidentală a continentului. 2 România: termen folosit pentru a desemna Imperiul roman în opoziţie cu lumea ba
utilizat constant în limbajul curent, dar nu şi în documentele oficiale, începând din secolul al IV-lea, pentru Imperiul roman şi, ulterior, pentru teritoriile biza

Introducere

Roma, al cărei geniu specific era cel al ordinii şi al tehnicii, al organizării şi al dreptului, dar care asimilase şi cultura şi arta Greciei, a sfârşit prin a face di

Figure 7. Editing the automatically extracted text.

After performing all the document processing steps, the book is saved and available to be accessed by users. Librarians can later on continue with editing the contents of the book and its metadata, while regular users can access its contents, perform searches and access the original PDF document.

Semantic Search

The search facility incorporated in the Lib2Life web portal relies on a semantic search algorithm. Due to the lack of annotations, the algorithm currently uses a K-Nearest Neighbors classifier with semantic distances based on word embeddings from ReaderBench. The workflow for semantic search is shown in Figure 8.

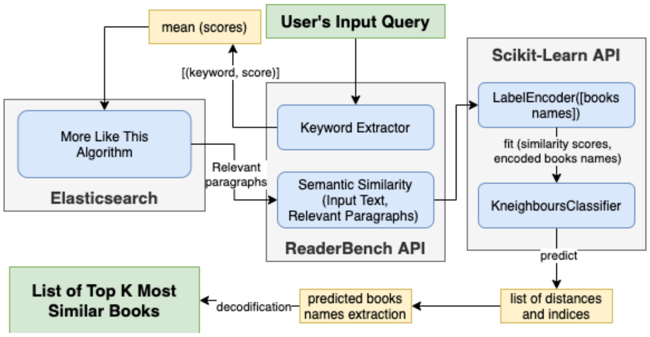

mean (scores)
User's Input Query
[(keyword, score)]
Scikit-Learn API
More Like This Algorithm
Elasticsearch
Keyword Extractor
Relevant paragraphs
Semantic Similarity (Input Text, Relevant Paragraphs)
ReaderBench API
LabelEncoder([books names])
fit (similarity scores, encoded books names)
KneighboursClassifier
predict
List of Top K Most Similar Books
decodification
predicted books names extraction
list of distances and indices

Figure 8. Semantic search algorithm.

The service extracts keywords from the input query text using the Keywords Extraction endpoint provided by the ReaderBench framework. The output consists of lemmatized content words, with their corresponding relevance score. This set of words is used to find the top k-nearest documents using the "More Like This" query incorporated in Elasticsearch. The result of this query consists of a list of paragraphs with at least one of the keywords. The similarity between the paragraph and the query embeddings is afterwards used to predict the top nearest documents. The corresponding document for each paragraph is retrieved, and a list with the most similar documents is returned (see Figure 9).

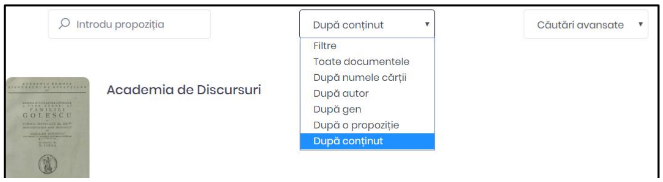


Figure 9. The semantic search functionality besides traditional search mechanisms.

The Web Portal

The Lib2Life platform provides a web interface developed for librarians and readers – see Figure 10. The User Interface (UI) is created using the Angular framework, version 9 (https://angular.io). Specific functionalities were developed for the two main roles: administrators (librarians) and readers. Librarians are able to upload, modify, and delete documents, while readers are only able to access the documents' content. Both librarians and readers can use the keyword-based search with filtering criteria, as well as the semantic similarity function to retrieve similar documents.

The web UI is connected to two application servers, one on Java, and another using the Flask framework developed in Python (https://github.com/pallets/flask). Both servers interact with an Elasticsearch instance for storing and retrieving indexed data. The ReaderBench API is used for performing advanced NLP processing, both for semantic search, and also for text extractions.

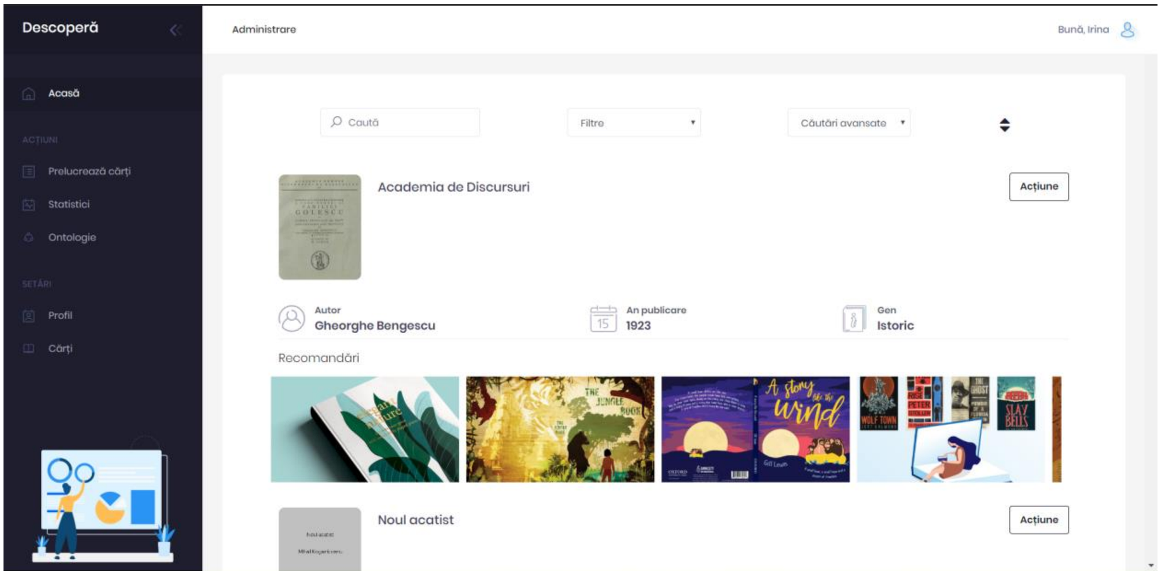


Figure 10. The Lib2Life dashboard.

The web portal incorporates an ontology viewer based on Protégé [10] – WebVOWL (http://vowl.visualdataweb.org/webvowl.html). The underlying ontology is used for text categorization and for better contextualizing the covered domains [7]. Figure 11 introduces the Lib2Life ontology viewer depicting a knowledge domain, namely Linguistics and Philology together with its corresponding subclasses.

RESULTS

A survey was conducted, and a questionnaire was distributed to evaluate the initial version of our platform. Twenty-three users aged between 20 and 50 years old, with a background ranging from students to Ph.D., working in a wide range of activity domains, were asked various questions about the platform and their experience with it. Demographic data showed that 54% of the respondents were aged 20-30, 21%

were aged 30-40, while 20% were aged higher than 40 years old. In terms of gender, 59% of users were women, while 41% were men. The education background included: high school – 4%, bachelor – 17%, master's degree – 54%, and Ph.D. – 25%. The activity domain ranged from IT in general – 34%, research in NLP – 29%, education – 29%, medicine – 4%, and graphic design – 4%. The users had to answer 14 Likert scale (1 – strongly disagree; 5 – fully agree) questions, which are presented in Table 1. Users found the web application intuitive, easy to use, and with a pleasing design. The questionnaire included four open-ended questions, allowing users to write opinions in natural language about what they liked or disliked, what features they missed, and what improvements should be performed to the application.

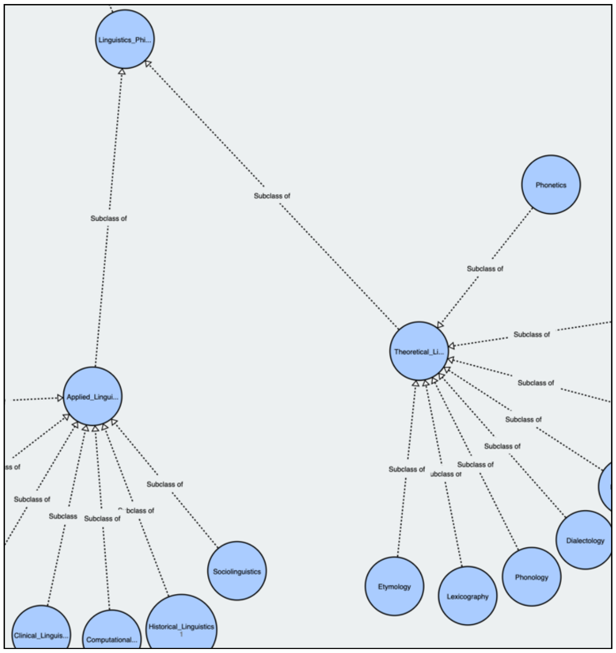


Figure 11. The Lib2Life ontology viewer.

Based on the answers to the open-ended questions, we found that the full document visualization page needs to be modified, both in terms of performance and of functionality. Users also requested having access to additional filtering criteria for browsing the collection of books. In addition, users encountered some errors and requested optimizations in terms of response time. Moreover, the implementation of a personalized bookshelf was also suggested.

A system limitation was met when text extraction algorithms did not work properly for some of the scanned documents. The solution relied on iteratively improving the OCR process. However, issues caused by improper scanning could not be always addressed.

In addition, the Lib2Life system did not differentiate amongst document types. Using another iterative process, the text extraction algorithm had to be constantly adapted based on characteristics shown by each new analyzed document. However, different categories or domains of documents may exhibit category-specific characteristics. Thus, we will consider improving the text extraction process by taking into account the domain of the document.

CONCLUSIONS AND FUTURE WORK

The Lib2Life platform aims to empower virtual restoration of historical documents owned by Central University Libraries in Romania by providing access to the digitalized documents in an online environment. Lib2Life currently stores about 100 documents provided by partner libraries which were manually corrected and used for testing.

Table 1. Feedback Questions.

#	Question	M (SD)
1	The Lib2Life application is intuitive.	4.57 (0.51)
2	The Lib2Life application is easy to use.	4.87 (0.34)
3	I could use the Lib2Life application without encountering errors.	4.13 (0.97)
4	The uploading and correction processes are easy to use.	4.52 (0.68)
5	The uploading and correction processes cover all the necessities.	4.43 (0.68)
6	The documents' visualization page is intuitive.	4.74 (0.69)
7	The search engine and the filters work as expected.	4.22 (0.80)
8	The search engine and the filters are intuitive.	4.22 (0.80)
9	The information included in the statistics page is useful.	4.90 (0.30)
10	The ontology is useful.	4.87 (0.34)
11	The Lib2Life application has an intuitive design and is suited to a system dedicated to libraries.	4.82 (0.50)
12	I would like to see more statistics about available documents.	4.43 (.843)
13	I would like to have more filters at my disposal on the dashboard page.	4.57 (.788)
14	I would like to be able to access documents from multiple domains and multiple languages.	4.91 (.288)

Lib2Life is a novel platform that includes useful filters for searching books, as well as the opportunity to read a document in PDF format. Users can also explore the entire domain ontology. The semantic search algorithm allows readers to find the most relevant documents for a query, or to be provided with similar documents to a selected one.

A usability questionnaire distributed to multiple users showed that the application is useful and includes a convenient semantic search functionality. The questionnaire argued that Lib2Life stands as a suitable software application to enable individuals to access digitalized historical documents. However, users requested improved response times, reducing error messages, and fixing server-related issues. In addition, a simpler user interface, but with more in-depth search criteria was also suggested.

The particularities of the old Romanian language used in the indexed historical documents should be also further explored. Namely, archaic words and structures should be considered in ReaderBench as the current version only supports contemporary language. This can be performed with the help of the eDTLR dictionary [3], which is an electronic Romanian dictionary containing more than 175,000 words. The temporal dimension should also be taken into account in a future version of the system, namely understanding how user preferences evolve over time, and followed by adjusting their recommendations.

Lib2Life enables librarians to build up a repository for their collection of documents. The resulting collection may be used for performing analyses focused on the evolution of literature across time. Example analyses include correlations to major historical events, changes in writing styles [5], as well as exploring inter-textual links between documents.

ACKNOWLEDGMENTS

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI - UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0689 / „Lib2Life - Revitalizarea bibliotecilor si a patrimoniului cultural prin tehnologii avansate” / "Revitalizing Libraries and Cultural Heritage through Advanced Technologies", within PNCDI III.

REFERENCES

1. Alewaeters, G., 1982. VUBIS: A user-friendly online system. *Information Technology and Libraries 1*, 3, 206-221.
2. Blei, D.M., Ng, A.Y., and Jordan, M.I., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3*, 4-5, 993–1022.
3. Cristea, D., Răschip, M., Forăscu, C., Haja, G., Florescu, C., Aldea, B., and Dănilă, E., 2007. The Digital Form of the Thesaurus Dictionary of the Romanian Language. In Proceedings of the 4th International IEEE Conference SpeDIEEE, 195-206.
4. Crossley, S.A., Dascalu, M., and McNamara, D.S., 2017. How important is size? An Investigation of Corpus Size and Meaning in both Latent Semantic Analysis and Latent Dirichlet Allocation. In Proceedings of the 30th Int. Florida Artificial Intelligence Research Society Conf. (Marco Island, FL), AAAI, 293–296.
5. Gifu, D., Dascalu, M., Trausan-Matu, S., and Allen, L.K., 2016. Time Evolution of Writing Styles in Romanian Language. In Proceedings of the 28th Int. Conf. on Tools with Artificial Intelligence (ICTAI 2016) (San Jose, CA), IEEE, 1048–1054.
6. Gormley, C. and Tong, Z., 2015. *Elasticsearch: The definitive guide: A distributed real-time search and analytics engine*. O'Reilly Media, Inc.
7. Gutu-Robu, G., Ruseti, S., Tomescu, S.-A., Dascalu, M., and Trausan-Matu, S., 2020. Designing an Ontology for Knowledge-Based Processing in Romanian University Libraries. In Proceedings of the The 16th International Scientific Conference eLearning and Software for Education (Bucharest).
8. Gutu-Robu, G., Sirbu, M.-D., Paraschiv, I.C., Dascalu, M., Dessus, P., and Trausan-Matu, S., 2018. Liftoff - ReaderBench introduces new online functionalities. *Romanian Journal of Human - Computer Interaction 11*, 1, 76–91.
9. Hämäläinen, W. and Vinni, M., 2006. Comparison of machine learning methods for intelligent tutoring systems. In Proceedings of the Int. Conf. in Intelligent Tutoring Systems (Jhongli, Taiwan), Springer, 525–534.
10. Knublauch, H., Fergerson, R.W., Noy, N.F., and Musen, M.A., 2004. The Protégé OWL plugin: An open development environment for semantic web applications. In Proceedings of the International Semantic Web ConferenceSpringer, 229–243.
11. Mikolov, T., Chen, K., Corrado, G., and Dean, J., 2013. Efficient Estimation of Word Representation in Vector Space. In Proceedings of the Workshop at ICLR (Scottsdale, AZ).
12. Nitu, M., Dascalu, M., Dascalu, M.-I., Cotet, T.-M., and Tomescu, S., 2019. Reconstructing Scanned Documents for Full-text Indexing to Empower Digital Library Services. In Proceedings of the 12th Int. Workshop on Social and Personal Computing for Web-Supported Learning Communities (SPeL 2019) held in conjunction with the 18th Int. Conf. on Web-based Learning (ICWL 2019) (Magdeburg, Germany), Springer, 183–190.
13. Rana, C. and Jain, S.K., 2012. Building a Book Recommender system using time based content filtering. *WSEAS Transactions on Computers 11*, 2, 27-33.
14. Sebastiani, F., 2002. Machine learning in automated text categorization. *ACM Comput. Surv. 34*, 1, 1-47. DOI= http://dx.doi.org/10.1145/505282.505283.
15. Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R., and Walker, J.H., 2003. DSpace: An open source dynamic digital repository.
16. Smith, R., 2007. An overview of the Tesseract OCR engine. In Proceedings of the Ninth international conference on document analysis and recognition (ICDAR 2007)IEEE, 629-633.

Emerging Patterns in Romanian Literature and Interactive Visualizations based on the *General Dictionary of Romanian Literature*

Irina Toma, Laurentiu-Marian Neagu, Mihai Dascalu, Ștefan Trăușan-Matu
University Politehnica of Bucharest
313 Splaiul Independentei, 060042, Bucharest, Romania
{irina.toma, laurentiu.neagu, mihai.dascalu, stefan.trausan}@upb.ro

Laurențiu Hanganu, Eugen Simion
The "G. Călinescu" Institute of Literary History and Theory, Romanian Academy
Calea 13 Septembrie, Bucharest, Romania
{institutulcalinescu, eugen.ioan.simion}@gmail.com

DOI: 10.37789/rochi.2020.1.1.4

ABSTRACT

The General Dictionary of Romanian Literature (DGLR) is a comprehensive work carried out by researchers from the literary institute of the Romanian Academy. DGLR offers detailed information about writers, editors, translators, literary publications, and cultural institutions that contributed to the Romanian national literature. The current work presents interactive web visualizations, based on statistical studies performed on the DGLR corpus, such as: biographical information; geographic information, including countries that part of the most important writers have visited, lived in, or studied in; active literary entities per year; timeline of publications for important writers. The purpose of the visualizations is to provide overviews regarding the Romanian literature to the general audience. In addition, our views offer valuable insights on the writers and their work across time. A survey was conducted on 20 users and most of them had a pleasant experience; recommendations on future developments were also provided.

Author Keywords

Analytical approach; Quantitative study; DGLR; Romanian writer; ReaderBench framework; Interactive visualizations.

ACM Classification Keywords

H.5.2: Information interfaces and presentation (e.g., HCI): User Interfaces;

I.2.7 Natural Language Processing: Discourse, Language parsing and understanding, Text analysis.

General Terms

Text analysis

INTRODUCTION

The ongoing work of the Romanian Academy for the national literature digitalization includes two important projects: a) the General Dictionary of Romanian Literature (DGLR), which contains information on the representative writers and institutions that contributed to national literature, and b) the Chronology of Romanian Literary Life (CVLR), which maps the relevant Romanian literary events between 1944 and 2000. The corresponding works are available through two channels, namely in printed form and in the INTELIT web platform. Several statistical analyses were performed using DGLR and CVLR corpuses, and corresponding results were integrated into the ReaderBench framework[1] [10].

The current work presents interactive web visualizations of writers' statistics based on DGLR, their integration into the ReaderBench platform, together with a qualitative study on the visualizations' usefulness and ease of use. The visualizations provide a broad, general perspective on the Romanian literature, through various charts depicting key points of writers' lives and their writings.

The paper is structured as follows. The next section presents the state-of-the-art, highlighting similar studies and available types of visualizations. Next, the third section presents the corpus, together with data processing techniques and technologies used for storage, integration, and visualizations. The fourth section presents interpretations of the views, followed by an evaluation based on a questionnaire. The last section draws the

[1] http://readerbench.com/

conclusions from the current analysis and outlines possible directions of future development.

STATE OF THE ART

Most analyses on literature are based on analytical approaches. For example, Moretti [15] introduces quantitative studies on the evolution and morphology of novels throughout history, including: visualizations for quantitative history (e.g., number of new novels per year), maps for cultural mapping (e.g., the protagonists of Parisian novels and the objects of their desire), and trees to represent evolutionary theory (e.g., evaluating the presence of clues in the early stages of British detective fiction).

The direction of his work was continued and enhanced in the Stanford Literary Lab [1, 19] via automatic tools of digital text analysis [16]. Moretti [14] also published a more comprehensive collection of essays that analyzes the morphological transformations in European novels, accompanied by research on novels' plots using network theory. His essays present quantitative information on: a) geography, as a fundamental factor in the divergence of different literary genders; b) the representation of character relations in plots, and c) statistical information on the titles, such as length or the use of adjectives and of proper names in titles.

The study by Sinykin et al. [18] was influenced by Moretti and it addresses the subjects of economics and race in American postwar novels. The study showed that women use 20% fewer economic terms than men, while African Americans use 10-15% fewer economic terms than Caucasian writers. Other studies analyze the link between book genre and writer gender in various types of writings [20], or apply cluster analysis on English novels to identify similarities in authors' writing style [7]. More in-depth studies, like the one introduced by Bode [2], analyze the evolution of the Australian novel from 1830 to the present days, the influences of other literatures, and the impact of women novelists in the national literature. The study displays statistical data, such as the number of novels by writer gender, top book publishers, places of publication, forms of publication, publisher category, genre of novels, topmost critically discussed writers etc.

The representations corresponding to the previous quantitative studies used classic visualization, such as line charts, bar charts, or manually drawn maps. Campbell et al. [4] proposed a modern representation of a collection of texts – Women Writers Online (WWO) [8] – to facilitate close and distant reading [11], and to provide easy access to general users. The WWO corpus contains more than 420 English texts written by women between 1626 and 1850, covering a wide range of topics and genres. Data representation consists of a bipartite network visualization that connects named entities to corresponding texts in which mentions can be found.

Jockers and Mimno [12] propose another modern visualization to identify how writer gender, nationality, and date of publication impact the theme of novels from the 19th century. The writers used a corpus of 3,346 novels from the 19th century covering British, Irish, and American fiction. The study is centered on identifying differences between male and female authors who write on various themes, such as: religion, war, or fashion. Moreover, the study also introduced an automated prediction of the gender of anonymous writers based on the previously generated model.

The current study is aligned with previous analyses by providing valuable insights on writers described in the DGLR through interactive visualizations. User have access in an interactive web platform to biographical information, geographical information (e.g., the countries the most important writers visited), literary entities, and publication timelines for the most important writers.

METHOD

Our solution is a web platform that integrates several visualizations of statistical data related to Romanian writers, their writings, and other literary entities, all corresponding to letters A-O from DGLR that were currently available. The targeted writings cover domains from folklore to literary theory and expand to writings from the Republic of Moldova to writings by German, Greek, or Jewish writers on the Romanian soil. Besides writings and writers, the dictionary also includes information about editors, translators, publicists, cultural and literary movements and concepts, magazines, and cultural institutions from Romania and from Romanian diaspora, as well as anonymous writings [3]. The second edition of the dictionary is now under development; it will be available in 8 volumes, covering the information in alphabetical order. Currently, 5 volumes belonging to the second edition are already published, including letters A–O.

The data used in our visual representations was automatically extracted from DGLR and from a set of files provided separately by the Romanian Academy; these files included more detailed, granular chronological information on the life of canonical writers. A first experiment by Neagu et al. [17] was conducted on a subset of the available corpus to observe demographics of Romanian writers based on DGLR. The current work follows the same direction, processing a larger amount of data, extracting additional types of entities, and introducing novel visualizations. Additionally, the views are integrated into the ReaderBench framework and are available to the general public, as presented below.

Indexing and Data Extraction

The corpus includes pre-print versions of DGLR together with a set of Microsoft Word documents that contain the chronology of life and literary activity of canonical writers (i.e., the most representative writers from Romanian

Literature). The information from the DGLR volumes is provided in Adobe InDesign[2] format, which was then converted to HTML for a standardized processing of data. The same process was applied to the Microsoft Word files which were converted to the HTML format.

The available DGLR corpus includes 2529 entities recognized as writers, from which 2433 authors were chosen for our work. The selection criterion consists of a valid year of birth that can be extracted from the description field. In addition to writers, 1186 entries were labeled as other entities (publications, associations, institutions, genres, etc.), and 1075 were included in our analysis. For the selected entries, the year of birth was found in the first line of their description using the common format "YYYY". Disregarded entries did not have a birth year associated – for example, genres specifying only the century ("appeared in the XVIII century"). In contrast to DGLR, the corpus for the chronology of life and literary activities included only canonical writers: "Lucian Blaga", "George Bacovia", "Mircea Eliade", "Constantin Dobrogeanu-Gherea", "George Coșbuc", "Ion Barbu", "Tudor Arghezi", "Mihai Eminescu", "Emil Cioran", and others.

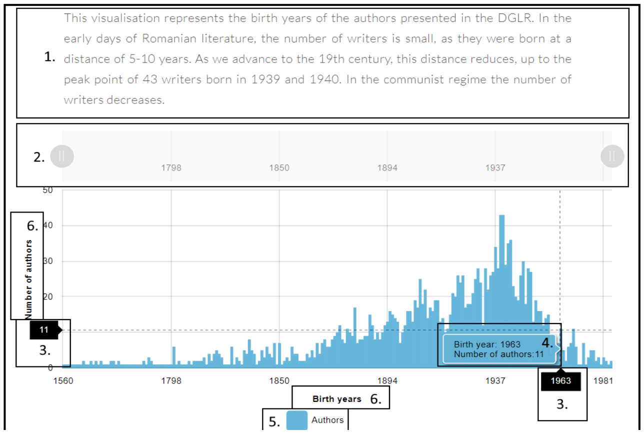
Data was stored in an Elasticsearch instance, suitable for analytics purposes and fast on data retrieval in large amount of texts [9]. Two indexes were created, *index-writers* and *index-publications*, respectively, to make a separation between each category as the data stored for each entity was different. The following fields for writers were extracted from DGLR: name, year and birthplace, year, and place of death (if it is the case), professions, text biography, publishing years, list of publications and critical references. For the other literary entities, we only extracted their name and the description. Specific data preprocessing techniques were performed to extract relevant information and to structure it accordingly, as required by the graphical tools.

Visualizations

AmCharts, a modern JavaScript library, was used to represent data. AmCharts can plot different types of views, such as: line, bar, or pie charts, as well as more complex views, such as maps, timelines, or Scalable Vector Graphics (SVG) pictorials. Besides the wide variety of views, AmCharts can render visualizations either from JSON inputs, or programmatically using the API for JavaScript or TypeScript. This was a major advantage for our application, as the standard visualizations were created using a JSON format.

The visual elements integrated in the standard charts are independent of the displayed data. As seen in Figure 1, each visualization is composed of:

- A detailed description that is displayed in the upper part of the page (1);
- A "smart" scrollbar that displays a miniature of the horizontal axis, together with two draggable bullets on each side of the scrollbar used for filtering the timeline (2);
- A cursor for better visualizing the values on the axis (3), together with tooltips available on hover for all the datapoints (4);
- A legend for each data series displayed in the chart (5);
- Labeled horizontal and vertical axis (6).



1. This visualisation represents the birth years of the authors presented in the DGLR. In the early days of Romanian literature, the number of writers is small, as they were born at a distance of 5-10 years. As we advance to the 19th century, this distance reduces, up to the peak point of 43 writers born in 1939 and 1940. In the communist regime the number of writers decreases.

Birth year: 1963
Number of authors: 11

Figure 1. Anatomy of a standard chart.

Geographic maps were implemented as separate Angular components, using the AmCharts API and the geodata package[3]. This package contains representations of the world countries in a GeoJSON [3] format. Each map comes in two resolutions, low and high, the difference between them being the number of points used for drawing the borders. The current visualizations use the low-resolution maps, as these are faster to load. In addition, an exact border representation is not vital for our charts.

ReaderBench Website

The ReaderBench website showcases tools for Natural Language Processing, Cohesion Network Analysis and text mining [6]. The website is developed in Angular[4] and is composed of numerous sections. The newly introduced visualizations were created in a separate page of the *ReaderBench* website, *Experiments*, centered on standalone analyses. The visualizations introduced in this paper are publicly available online, free of charge, at http://readerbench.com/experiments/intellit.

[2] https://www.adobe.com/products/indesign.html

[3] https://www.npmjs.com/package/@amcharts/amcharts4-geodata

[4] https://angular.io/

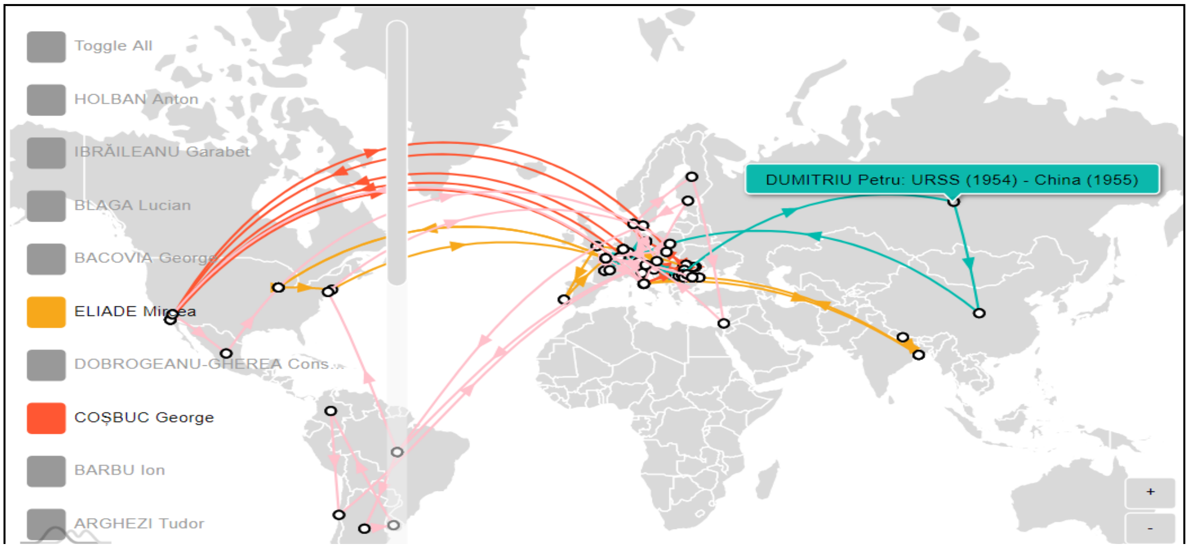


Figure 2. The travels of canonical writers.

Experiments

The information extracted from DGLR is available to the end users as graphical visualizations divided into three categories, based on the represented data:

- *Writers throughout time* – this category contains the number of writers born each year, the birth location of the writers represented on a map, the death age of the writers, the number of publications and active writers per year;
- *Literary entities throughout time* – two views are considered, depicting the number of publications per year and the number of active publications per year;
- *Canonical writers' life* – in this category, we display the cities visited by canonical writers, their active publication period, as well as a publication timeline.

The representations can be separated into the following categories based on their type: bar-charts, maps, and miscellaneous. As a standard, the bar-charts from this experiment display on the horizontal axis the timeframe between 1515 and 2010, corresponding to the first and last recorded publication years in the DGLR. The vertical axis displays different values, depending on the selected view: the number of writers born that year, the number of publications issued, the number of active publications, or the number of active writers together with the number of published works. Another observation regarding this type of views is the existence of spikes or gaps. A line indicator depicting a 5-point moving average was applied to better highlight the trends and smooth the evolutions by filtering short-term fluctuations.

The second type of visualizations consists of two maps: the writers' birth places and the travels of canonicals writers throughout the world. These types of visualizations were introduced in the initial study performed by Neagu et al. [17], but the travel map of canonical writers was enhanced, as follows (see Figure 2). The displayed paths can now be filtered by selecting and deselecting entities from the left-side legend. The "Toggle all" button removes or adds all writers from/to the map. Path directions were introduced for all travel segments, together with a tooltip displaying the start and end locations. In addition, buttons for finer zoom control were added to the bottom-right corner of the screen.

The last category of visualizations, miscellaneous, contains three visualizations. The first view is an area chart displaying the death years of writers (see Figure 3). Each year in the chart has three corresponding values: the death age the youngest and oldest authors, together with the average value between these two. Second, a miscellaneous graph considers a dumbbell plot for the publications of canonical writers [17], listing the first and last publication years. Third, we introduce an experimental timeline view, an alternative to the canonical writers' travels, available currently only for "George Coșbuc" (see Figure 6), a representative writer for Romanian literature; other authors will be added iteratively to this visualization. Each section from the timeline view is colored differently, corresponding to a time period and place where the author travelled to; the name of the place is displayed on mouse hover. The writer's publications are displayed chronographically, colored for consistency similarly to the corresponding period.

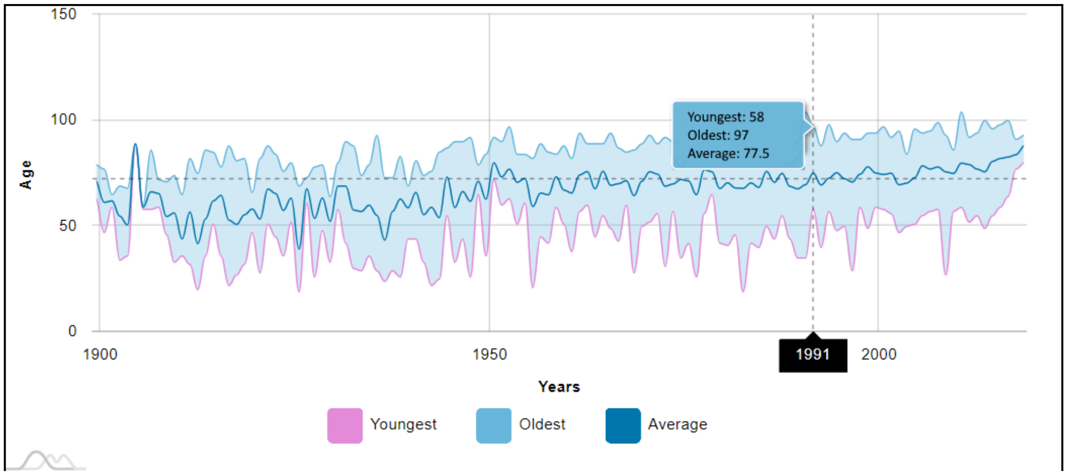


Figure 3. Age of death for writers.

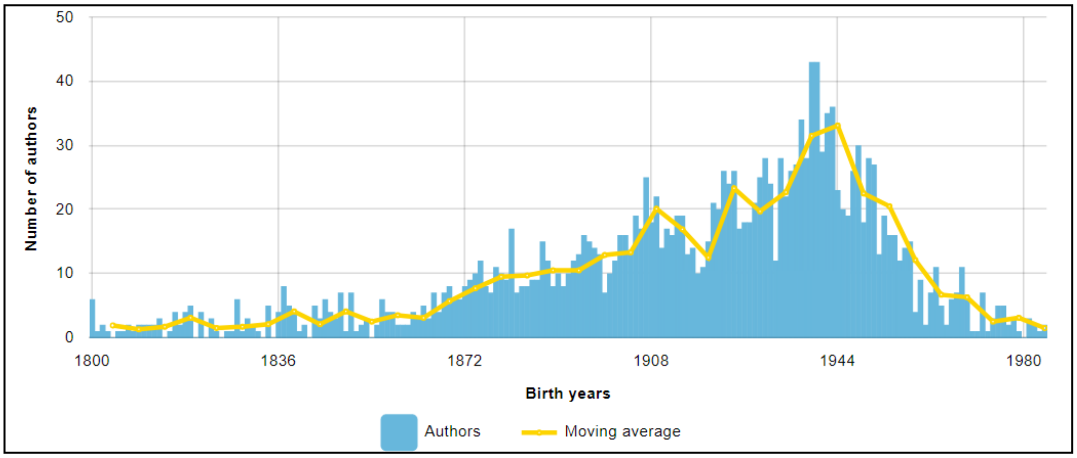


Figure 4. Writers born per year.

DISCUSSIONS

Our visualizations are grouped into three main categories: writers throughout time, literary entities throughout time, and the life of canonical writers. Each category has a list of visualizations presented below.

Romanian Writers throughout Time

The first visualization in this category aims to depict the age of the youngest and oldest writers who died every year, and the average writers' age in the 20th and 21st centuries (see Figure 3 depicting the lowest, highest, and average value).There were many consecutive years between the 16th and 19th centuries for which there was no available data, or several years when only one writer was present. Writers before 20th century were not included in this plot. Up until 1809, only 3 years marked the death of at least 2 writers: 1711, 1715, and 1724. Data is less sparse between 1809 and 1900, but the plot would have included a lot of gaps due to the low number of writers in that period; more data was available after 1900 and until 2018.

The oldest writers in Romanian literature had 104 years (two writers), close to them one writer died at 101 years, other three at 100 years, two at 99 years, and some others at 98. At the lower end, the youngest writers who died after 1900 had only 19 years (two writers), one 20 years, and two writers were only 22. The average age of Romanian writers is 68.89 years for the full historical period included in our dataset.

The second visualization in this category (see Figure 4) highlights that the number of writers is small in the early days of Romanian literature (maximum 6), as they were born at a distance of 5–10 years; hence, data before 1800 was excluded from the analysis. The distance between the writers' years of birth reduces starting from 1800, as we advance in the 19th century. The peak is the year 1881, when 17 writers were born; close to it are 1895 (16 writers were born) and 1887 (15 writers were born).

The observed pattern is that the Romanian literary contributions started to intensify from the middle of the 19th century. Afterwards, we observed that at least one writer was born each year in the period 1900–1980, with the peak in 1939 and 1940 (when 43 writers were born each year), a very challenging period worldwide marking the beginning of World War II. The most flourishing period in history for the birth of Romanian writers is 1920–1951, when 17 or more writers were born each year, except 1932. Afterwards, a fall in the number of born writers was observed in the communist regime. Nevertheless, the youngest writers alive are born in 1984 (currently 36 years old); this may show that there are still future writers which are not yet well known and may fill in these gaps.

Romanian Literary Entities throughout Time

The first visualization in this category (see Figure 5) is related to the literary entities extracted from DGLR: literary publications, associations, and cultural institutions. Results are interesting by highlighting that the interwar period was most flourishing for the Romanian literature. Also, an important spike is shown in 1990, immediately after the communism fall, when the largest number of literary entities was encountered.

Another analysis in this category presents the literary entities active per year. The start and end years were considered the same for literary entities which had only the start year in the dictionary. There are entities which were active during certain periods of time and had missing years of activity due to wars or other internal financial problems.

Canonical Writers Life

A visualization in this category includes an interactive timeline of a writer's literary activity. Figure 6 presents the timeline chart for "George Coşbuc", a well-known Romanian writer. The timeline displays each work of the author: the work name extracted from DGLR, alongside with its corresponding publication year, together with the location where it was written.

Another analysis includes the cities visited by the canonical writers with their corresponding years (see Figure 2). This visualization shows a world map with arrows drawn between start and end cities, alongside tooltips with corresponding details.

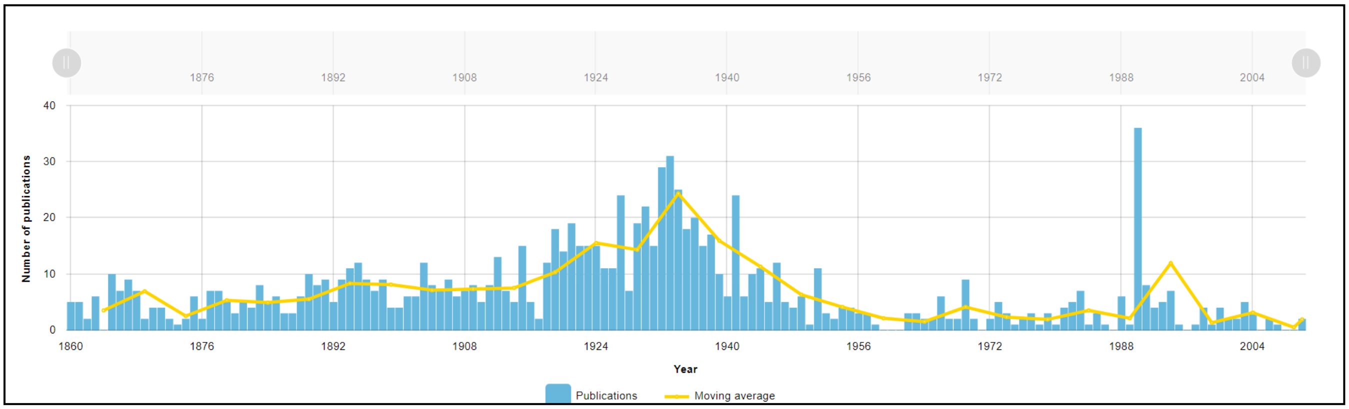


Figure 5. The number of literary entities born per year.

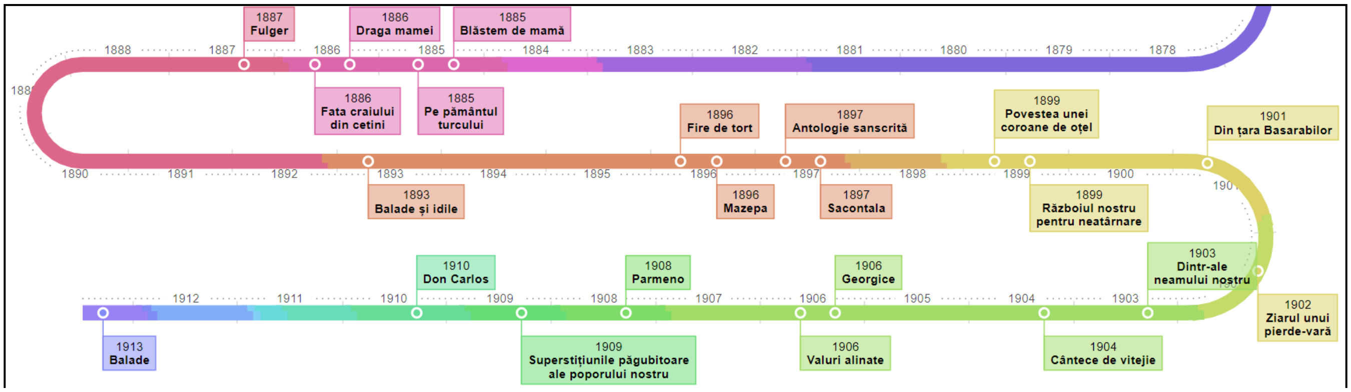


Figure 6. Timeline chart for George Coşbuc.

USER TESTING

A survey was conducted on 20 users, 15 males and 5 females, with ages between 25 and 45 years old. All users were asked to respond to a survey with 6 questions having ratings on a 5-point Likert Scale (1 indicates complete disagreement and 5 complete agreement) followed by 4 free-input questions that cover their opinion on the interface and the functionalities. The first 6 questions targeted the ease of use for the more complex visualizations (i.e., canonical writers' travels, timeline, writers' age of death), the overall quality of the information in the interface and the usefulness of the moving average. User were asked in their open-ended questions to provide feedback regarding the view concerning the writers' birth locations, their favorite visualization, and describe what type of information and visualizations they considered most valuable.

Two reliability statistics were calculated for the recorded answers. The Intraclass Correlation Coefficient (ICC) [13] of .624 and Cronbach's Alpha [5] of .778 denote a moderate level of agreement between the users.

The feedback received from the free-input questions was comprehensive. The first question considered the users' opinion regarding the birth location of writers represented on the map. Eighty percent of the users considered the representation interesting and useful, and 11 users requested more information on the map, such as the names of the writers and the birth years. Also, another interesting suggestion was to filter the map and display information for certain periods of time. Most users complained about the information which was too cluttered, and they experienced rendering issues. Improvements were also suggested, for example: displaying only the writers born in Romania and aggregating the rest of the writers per country. The second suggestion was to color the counties based on the number of writers born in each of them, generating a heatmap.

The second question requested users to point their favorite view and argument their choice. The results are presented in Table 1. Three users selected two views, while one user checked all visualizations as favorites. The maps and the timeline views were considered most attractive and interactive. The other charts were preferred by users who went beyond the raw information and correlated the values with historical events.

Table 1. Number of votes per type of view.

Favorite view	Number of votes
Canonical writers' travels	8
Timeline for George Coșbuc	6
Writers' birth years	3
Writers' birth places	3
Canonical writers' publications	1
Publications and active writers	1
All	1

The next two questions covered new information or visualizations that the users wanted to see in the interface. We received 22 different ideas, and we will focus on the most frequently recurring suggestions:

- Adding a visualization that represents the literary movements and the most representative corresponding writers *–five votes*;
- Extending the timeline to more writers *– four votes*;
- Introducing a tutorial for interacting with the visualizations *–three votes*;
- Adding the name of the writers/publications in views that support this feature *– three votes*;
- Depicting how events at worldwide or personal levels affected the works of the writers *–two votes*;
- Adding a view with the most important publications – *two votes*.

CONCLUSIONS AND FUTURE WORK

The multilateral process of shifting the Romanian literature to the digital era involves literary researchers, linguists, and computer science specialists. The current paper aims to explore statistical information and web-based interactive visualizations to display data from the General Dictionary of Romanian Literature in a simple and clear way for the broad audience. The results of a survey show that most end users had a pleasant experience with our views. Future development recommendations were provided, which will be integrated in the next versions of the website.

Future work includes the integration of remaining letters from DGLR (letters P-Z), which are still under development. Moreover, the timeline view will be extended to all canonical writers. Based on the user feedback, we will address the rendering issues, add information about the writers' and other literary entities' names. Additional visualizations are envisioned, such as a heatmap for the birth places of writers, a map of the death locations of the writers, marking worldwide events on the bar-charts, as well as a short tutorial for interacting with the representations.

In terms of data sources, we plan to integrate external sources containing historical events and foreign authors, and to perform cross-correlations to observe how the social, political, and economic context influenced the Romanian writers.

ACKNOWLEDGMENTS

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 54PCCDI ⁄ 2018, INTELLIT – "Prezervarea și valorificarea patrimoniului literar românesc folosind soluții digitale inteligente pentru extragerea și sistematizarea de cunoștințe" and by the Operational Programme Human Capital of the Ministry of European Funds through the Financial Agreement 51675/09.07.2019, SMIS code 125125.

REFERENCES

1. Allison, S., Heuser, R., Matthew, J., Moretti, F. and Witmore, M. 2011. Quantitative Formalism: an Experiment. *Stanford Literary Lab*.
2. Bode, K. 2012. Reading by numbers: Recalibrating the literary field. Anthem Press.
3. Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S. and Schaub, T. 2016. The geojson format. *Internet Engineering Task Force (IETF)*. (2016).

4. Campbell, S., Yu, Z.-Y., Connell, S. and Dunne, C. 2018. Close and Distant Reading via Named Entity Network Visualization: A Case Study of Women Writers Online. *Proceedings of the 3rd Workshop on Visualization for the Digital Humanities. VIS4DH* (2018).
5. Cronbach, L.J. 1951. Coefficient alpha and the internal structure of tests. *Psychometrika*. 16, 3 (1951), 297–334.
6. Dascalu, M., Dessus, P., Trausan-Matu, S., Bianco, M. and Nardy, A. 2013. ReaderBench, an environment for analyzing text complexity and reading strategies. *16th Int. Conf. on Artificial Intelligence in Education (AIED 2013)* (Memphis, USA, 2013), 379–388.
7. Eder, M. 2017. Visualization in stylometry: Cluster analysis using networks. *Digital Scholarship in the Humanities*. 32, 1 (2017), 50–64.
8. Flanders, J. 2002. Learning, reading, and the problem of scale: using women writers online. *Pedagogy*. 2, 1 (2002), 49–59.
9. Gormley, C. and Tong, Z. 2015. Elasticsearch: The definitive guide: A distributed real-time search and analytics engine. O'Reilly Media, Inc.
10. Gutu-Robu, G., Sirbu, M.D., Paraschiv, I.C., Dascalu, M., Dessus, P. and Trausan-Matu, S. 2018. Liftoff - ReaderBench introduces new online functionalities. *Romanian Journal of Human - Computer Interaction*. 11, 1 (2018), 76–91.
11. Jänicke, S., Franzini, G., Cheema, M.F. and Scheuermann, G. 2015. On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges. *EuroVis (STARs)* (2015), 83–103.
12. Jockers, M.L. and Mimno, D. 2013. Significant themes in 19th-century literature. *Poetics*. 41, 6 (2013), 750–769.
13. Koch, G.G. 1982. Intraclass correlation coefficient. *Encyclopedia of Statistical Sciences*. S. Kotz and N.L. Johnson, eds. John Wiley & Sons. 213–217.
14. Moretti, F. 2013. *Distant reading*. Verso Books.
15. Moretti, F. 2005. Graphs, maps, trees: abstract models for a literary history. Verso.
16. Moretti, F. 2016. Literature, Measured. *Stanford Literary Lab*.
17. Neagu, L.M., Toma, I., Dascalu, M., Trausan-Matu, S., Hanganu, L. and Simion, E. 2020. A Quantitative Analysis of Romanian Writers' Demography based on the General Dictionary of Romanian Literature. *5th conference on Smart Learning Ecosystems and Regional Development (SLERD 2020)* (Bucharest, Romania, 2020).
18. Sinykin, D., So, R.J. and Young, J. 2019. Economics, Race, and the Postwar US Novel: A Quantitative Literary History. *American Literary History*. 31, 4 (2019), 775–804.
19. Stanford Literary Lab: *https://litlab.stanford.edu/pamphlets/*. Accessed: 2020-04-20.
20. Thelwall, M. 2017. Book genre and author gender: romance> paranormal-romance to autobiography> memoir. *Journal of the Association for Information Science and Technology*. 68, 5 (2017), 1212–1223.

Violence Detection in Images Using Deep Neural Networks

Edwin-Mark Grigore
University POLITEHNICA of Bucharest
Splaiul Independenței 313, București 060042
edwmrkg@gmail.com

DOI: 10.37789/rochi.2020.1.1.5

ABSTRACT

Software and technology has evolved and expanded so much over the last decades, that is present in everybody's life in every little aspect, and more and more significantly at children's disposal. Starting from this reality, it is necessary the identification of the images that contain scenes of violence or emotionally disturbing scenes, images that contain blood or depict human bodies with open wounds, violent fires, or presence of guns and weapons. Machine learning (ML) is capable of extracting features from images and learn to identify the images that depict inappropriate scenes for children, using different techniques. With the recent advances in deep learning, traditional ML methods, such as Support Vector Machines, have been surpassed by deep neural networks that are also employed by our solution for violence detection.

Author Keywords

Computer vision; violence; violence detection; neural network; deep learning.

ACM Classification Keywords

I.2.10: Vision and Scene Understanding

General Terms

Computer Vision; Deep Learning; Violence Detection.

INTRODUCTION

The common adage "A picture is worth a thousand words" denotes exactly how an image can influence a child, especially if we are talking about inappropriate images. Browne and Hamilton-Giachritsis [1] have shown that aggressive or antisocial behaviour is heightened in children after watching violent television or films. Early exposure to extremely fearful events affects the development of the brain, particularly in those areas involved in emotions and learning [2]. When children see images that are emotionally disturbing, images that depict the world in an inadequate manner for their young minds to comprehend, they can learn fear from situations they should not be exposed to.

In order to prevent the exposure of children to graphic and violent images, these images must be firstly identified. Since parents cannot be physically near their children every single time, nowadays they can rely on the technology they use to achieve this task.

In a World Health Organization report, Krug et al. define violence as "the intentional use of physical force or power, threatened or actual, against oneself, another person, or against a group or community, which either results in or has a high likelihood of resulting in injury, death, psychological harm, maldevelopment, or deprivation" [3].

Transposing the notions to the field of images, violence transcends these categories. The term explicit or graphic violence refer to depiction acts of violence in visual media such as film, television, and video games. The violence may be real or simulated.

Graphic violence generally consists of uncensored depiction of various violent acts which includes depiction of murder, assault with a deadly weapon, accidents which result in death or severe injury, and torture.

As the technology advances, Computer Vision leads the way in training artificial intelligence to learn to interpret and understand the visual world. Using deep learning models, we now have machines that can accurately identify and classify objects. Although there is extensive knowledge to develop such deep learning models, only a few have been created that recognize and/or classify the violence depicted in still images, and even less are available for public use.

There are several potential areas that may use machine learning to detect violence, such as parental control applications, and web filtering. Therefore this topic is worth being studied and relevant for computer-human interaction researchers and product designers.

Transfer Learning is used in our work to build a model that detects and classifies violence in still images. This method applies different existing models that have already been trained for general purposes, to the characteristics of the task at hand. First, we must choose from the large pool of the existing deep learning models one to be the basis for our solution. The process of identification of the model that works best on the dataset available is considered to be a key aspect. Second, we take the pre-trained model and use it as a starting point for our violence detection model.

Also, we must decide which layers of the pre-trained model are used in the process and what layers must be built on top of it. Finally, we must adapt and refine the model so it may fit as well as possible the task at hand, process called fine-tuning the model.

The rest of this paper is organized as follows. Section 2 presents the categories of violence that are detected by the proposed solution. Section 3 introduces the dataset used in training the model. Section 4 presents the approach for violence detection and summarizes the results. Section 5 concludes the paper.

CATEGORIES OF VIOLENCE

Our proposed machine learning solution is intended to identify the following classes. The following lines briefly describe them and how they are connected to violent graphics.

- **Presence of firearms** – the presence of any type of gun or similar fire weapon, whether it is shooting or not, pointed at someone, or threatening a person, regardless of the intent of the subject depicted, is to be classified as violent image.
- **Presence of cold weapons** – any type of melee weapon, ranged weapon or other type of weapon that does not involve fire or combustion is to be classified as violent image.
- **Presence of fire** – any explosion caused by a bomb, any large-scale fire, vegetation fire, any human or animal, living or dead that is burning, any fire caused by a gun is classified as violent.
- **Fight scenes** – any image that represent a fight, regardless of the number of people involved or how they are fighting or the weapons they use, is to be classified as violent image. A fight scene may imply punching, kicking, mutual or from one side. Battle scenes struggles between a person and an animal will also be included in this category.
- **Presence of blood and gory scenes** – any serious body injury, any presence of blood that drains out from a body, any wound or tissue damage, any dead body that shows significant injury, presence of horror creatures, mutant creatures or skull and flesh representation is classified as violent image.

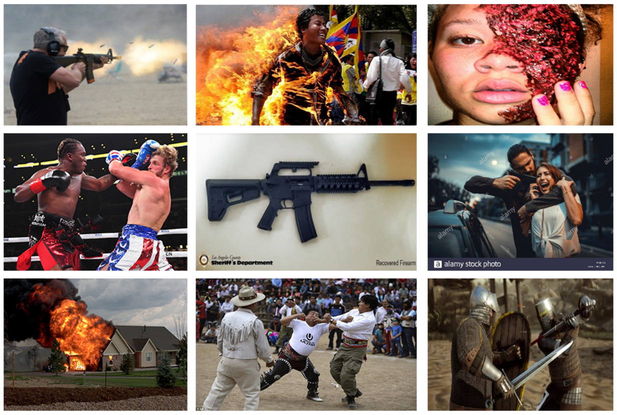


Figure 1. Example of violent images from the dataset

Any other image that is not classified under the above categories will be labelled as **non-violent**.

DATASET

A complete dataset is mandatory in order to train the model properly and to achieve good results. Due to the fact that only a handful of violence detection models have been proposed, we were unable to find an existing public database about violence in images with all the categories included. Consequently, we opted for a database of videos to start building our dataset.

Violent Scene Dataset (VSD) created by InterDigital [4-7] is a public dataset for the detection of violent scenes in videos. It is a collection of labels, features, and annotations based on the extraction of violent scenes from films and web videos. It also contains audio annotations of violence-depicting sounds present in the videos.

The dataset consists of 86 short videos downloaded from YouTube and normalised to a frame rate of 25. Also, the dataset contains ground truth created from a collection of 32 films of different genres (which are not included in the dataset due to copyright issues).

The violence identification is made based on two definitions of violent scenes: (1) subjective definition and (2) objective definition. The subjective definition describes a violent scene as a “scene one would not let an 8-year-old child see because they contain physical violence”[1]. The objective definition shows that a violent scene contains “physical violence or accident resulting in human injury or pain”.

Frames Extraction

Due to the fact the dataset contains videos, not images, processing work needed to be done. Each frame was extracted from the video, sorted according to the annotation and saved into the new database we created. At the end of the extraction, manual inspection of the resulting set of images was required. Duplicate images and images that are blurry, darkened or where the subject is unclear were removed. Also mislabelled images were moved to the proper category or removed if necessary. In the process of video and image manipulation we used the OpenCV library[2].

The number of images resulted in the process of extraction is in the tens of thousands. However, after a thorough manual inspection and repeated deletion of the unusable files, the database consisted of only around 1000 images, which is rather small for a machine learning solution. Also, different

[1] As presented in the description of the dataset. Available online at https://www.interdigital.com/data_sets/violent-scenes-dataset, last accessed 25 July 2020.

[2] http://opencv.org

perspectives on different categories failed to be gathered into the database, especially for gore and fire presence. In this case, we used Google image search to extend the dataset with graphics that picture the situations that were missing.

Augmentation

Because the dataset is small compared to what a proper dataset would look like, augmentation was helpful to extend the original database. Keras [8] interface allows us to augment the training set after loading the images in memory. It offers multiple ways to do the augmentation, such as image rotation, zooming, cropping, horizontal and vertical flipping, or range shifting.

Another recommended method is mixup [9] used as a regularization technique. Because we do not know the real distribution of data which can lead to overfitting, mixup comes into help to reduce this problem. It introduces combinations of pairs of images and their labels. A shallow explanation is given by this equation, where t is the ratio of mixing two images (a number between 0 and 1):

$$img_{new} = t * img_1 + (1-t) * img_2$$

SOLUTION AND EXPERIMENTAL RESULTS

In the process of implementation several well-known state-of-the-art technologies have been used. The model was developed and trained using TensorFlow [10].

In Computer Vision, deep learning has been used for tasks such as object identification or scenes recognition. Most solutions employ Convolutional Neural Networks, where lower layers act as feature extractors and the top layers work on the features that are specific to the task. Deep learning models learn different features on their architectural layers. These layers are often connected to a final fully connected layer to get the result. This layered architecture enables us to disconnect the final layer from the network and use the rest of the network as a feature extractor.

An important step in the successful training of the model is choosing the best neural model to apply transfer learning onto. All are state-of-the-art technology, but not all of them suit every problem. It is fundamental to have a model that offers good performance on the task it has been trained on.

In computer vision, several pre-trained neural models have been proposed in recent years. The resulting state-of-the-art deep learning networks are available to the large public and can be used freely and easily, both online or directly integrated in machine learning libraries.

The most popular and best performing such models are: VGG16 [11], InceptionV3 [12], Xception, and ResNet50 [13]. These are the ones we also considered using in developing the model for our task.

On top of these pre-trained models, we built a classifier using the weights from pre-training, consisting of a pooling layer, a few core neural layers and two normalization layers. The final layer is a fully connected layer with 6 (as the number of categories) neurons as output. For the Dense layers we used ReLU activation function, except for the output layer, where we used SoftMax. After splitting the dataset into training set and validation set of 75%-25%, we used the batches generated by Keras and trained the model for various epochs, ranging from 25 to 100. We employed RMSprop with a learning rate of 10-4 as optimizer.

In the process of training the models, the mixup technique helped to deal with overfitting, increasing the validation accuracy by 3-4%. The best performing models were ResNet50 and VGG16. They both provided similar results, but with variable epochs' number (see Table 1). The ResNet model peaks fast, reaching the top validation accuracy after only 13 epochs and maintaining it through the next epochs (up to 100), while VGG16 needs more training time to do so. In general, VGG16 required more time for training with the same batch and dataset size than ResNet50.

Table 1. Accuracy rates on training and validation sets

Epochs **Model**	**25**	**50**	**75**	**100**
ResNet50 without mixup				
Train. Acc.	79.89%	90.56%	91.63%	94.01%
Valid. Acc.	65.34%	71.59%	73.86%	70.85%
VGG16 without mixup				
Train. Acc.	70.31%	79.69%	84.94%	89.23%
Valid. Acc.	73.58%	77.56%	81.25%	80.68%
ResNet50 with mixup				
Train. Acc.	72.54%	82.58%	86.36 %	88.83%
Valid. Acc.	71.88%	65.62%	66.76%	71.02%
VGG16 with mixup				
Train. Acc.	64.11%	73.39%	78.50%	83.52%
Valid. Acc.	69.89%	76.99%	80.97%	80.68%

In Table 2, we can see a comparison between the performances of the models that have been trained with different feature extractors.

Table 2. Performance comparison for different neural model

Pretrained Model	**Accuracy**
InceptionV3	30%
Xception	41%
ResNet50	74%
VGG16	**81%**

Finally, the model we built using Transfer Learning based on the VGG16 pre-trained model is the one that performed the best, with an accuracy of 81%. Figure 2 shows examples of classification.

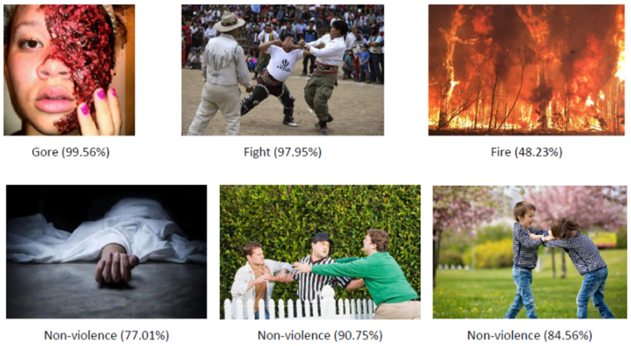


Figure 2. Examples of classification output

CONCLUSIONS

The outcome of this project shows that there is a lot to be done for improving the detection of violent images. Children can be protected using state-of-the-art computer vision technology and, by building a model that would detect the images that can be harmful to see for them and that classify the violence depicted in still images, we believe people can be encouraged to address this issue more. Building machine learning models and using them in all kinds of applications will, eventually, make the world safer for children.

Developing a deep learning model that recognizes violent scenes that would have an emotional impact over an 8-year old child by using deep neural networks is my proposal of work in this field. We built the model by aggregating the knowledge of a pre-trained model and a classification network, with VGG16 being the appropriate state-of-the-art model for the task. The model reports whether a violent or harmful scene is depicted in the image and outputs the class predicted and the score.

Future work will strive to increase the accuracy of the model. This can be acquired by gathering more data and by tuning the model better. Each class has its unique features and there is work to be done to refine the database of each violence category and to identify the features that will increase the accuracy of prediction. As the model will improve, it can be integrated in the parental application that will allow live detection of violence in accessed images.

ACKNOWLEDGEMENT

This work was supervised by Conf. Dr. Ing Traian-Eugen Rebedea, whose advices and guidance helped me in achieving the results.

REFERENCES

1. Browne, Kevin & Hamilton-Giachritsis, Catherine. (2005). *The influence of violent media on children and adolescents: A public-health approach*. Lancet. p. 8
2. National Scientific Council on the Developing Child. (2010). *Persistent Fear and Anxiety Can Affect Young Children's Learning and Development: Working Paper No. 9.* Retrieved from www.developingchild.harvard.edu
3. Krug et al., *World report on violence and health.* Archived 2015-08-22 at the Wayback Machine, World Health Organization, 2002.
4. C.H. Demarty, C. Penet, M. Soleymani, G. Gravier. (2014). *VSD, a public dataset for the detection of violent scenes in movies: design, annotation, analysis and evaluation.* In Multimedia Tools and Applications, May 2014.
5. C.H. Demarty, B. Ionescu, Y.G. Jiang, and C. Penet. (2014). *Benchmarking Violent Scenes Detection in movies.* In Proceedings of the 2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI), 2014.
6. M. Sjöberg, B. Ionescu, Y.G. Jiang, V.L. Quang, M. Schedl and C.H. Demarty. (2014). *The MediaEval 2014 Affect Task: Violent Scenes Detection*. In Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Spain (2014)
7. C.H. Demarty, C. Penet, G. Gravier and M. Soleymani. (2012). *A benchmarking campaign for the multimodal detection of violent scenes in movies*. In Proceedings of the 12th international conference on Computer Vision – Volume Part III (ECCV'12), Andrea Fusiello, Vittorio Murino, and Rita Cucchiara (Eds), Col. Part III. Springer Verlag, Berlin.
8. Keras | TensorFlow Core. TensorFlow. (2020). Retrieved 2020-06-08, from https://www.tensorflow.org/guide/keras.
9. Zhang, H., Cisse, M., Dauphin, Y., & Lopez-Paz, D. (2017). mixup: Beyond Empirical Risk Minimization. arXiv: abs/1710.09412
10. Metz, Cade (November 9, 2015). "Google Just Open Sourced TensorFlow, Its Artificial Intelligence Engine". Wired. Retrieved 2020-06-08, from https://www.wired.com/2015/11/google-open-sources-its-artificial-intelligence-engine/.
11. Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.
12. Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., & Anguelov, D. et al. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2015.7298594
13. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.90

Automatic detection of cyberbullying on social media platforms

Ștefăniță Stan
University Politehnica of Bucharest
Splaiul Independenței 313,
București 060042
st.stan96@gmail.com

Traian Rebedea
University Politehnica of Bucharest
Splaiul Independenței 313,
București 060042
traian.rebedea@cs.pub.ro

DOI: 10.37789/rochi.2020.1.1.6

ABSTRACT

The presence of cyberbullying on the Internet has grown alarmingly in recent years. Teenagers and children are the most affected by this phenomenon that is often the cause of higher suicide rates and social isolation. The detection and prevention of cyberbullying depends firstly on its correct understanding and secondly on the correct selection of a classification model trained on features that have a high discrimination factor between cyberbullying and non-cyberbullying. In this paper, we aim to create an automatic detection model for cyberbullying posts that is not biased towards a specific social media platform or a certain type of bullying. We describe the method we used for selecting the best features for two different classifiers trained on datasets collected from Twitter and Formspring. Next, we explain how we use the predictions made by these classifiers for labelling a new dataset collected by us from Twitter. The results of the automatic classification of the dataset have been compared to the manual classification of a sample of data from it, resulting in a rate of agreement larger than 50% between automatic detection and human annotation.

Author Keywords

Cyberbullying detection; Natural language processing; Social media; Online Aggressivity.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. .

General Terms

Human Factors; Languages; Measurement.

INTRODUCTION

Bullying is an increasingly frequent phenomenon, making its way into the Internet space, in the form of cyberbullying, along with the rising popularity of social media platforms. During a campaign meant to fight cyberbullying led by Bitdefender in 2017, *NU TASTA URA* (In English, *DON'T SEND HATE*)[1], an analysis of the cyberbullying phenomenon was made by questioning teenagers that were part of the Facebook group *Offtopic*, infamous for aggressive interactions between its members. The results have shown that about 80% of the questioned teenagers were victims of some level of cyberbullying (private or public), 66% of which did not tell anyone about it or asked for help. Moreover, only 36% of teenagers that witnessed instances of cyberbullying reported it or did something to help the victim. One of the characteristics of this phenomenon that makes it harder to notice than regular bullying is that it usually happens online, where parents or teachers may not see it. Also, its persistent and permanent nature (cyberbullying usually happening over a long period of time and instances of it remaining forever on the Internet space) have the capability of destroying the reputation of the aggressor, as well as of the victim. Therefore, research efforts were put into detecting and solving cyberbullying as quickly as possible in order to prevent the issues that may arise from it.

Most studies regarding this topic were done in the social and psychological fields to understand how this behavior appears, as well as how it affects both victims and bullies. Only recently it became a subject of interest for the information technology community, research being made on social media networks in order to find solutions for quick detection and prevention of cyberbullying. What scientists quickly discovered was that several issues arise when trying the traditional methods of text classification on this problem. The main issue is that there is not a consensus regarding a clear definition of cyberbullying, thus different researchers use different definitions, making future work harder, as there is not a solid basis for comparison. However, most of the definitions make references to the persistent and aggressive nature of cyberbullying. In this paper we use the definition given by Smith et al. [1] that defines cyberbullying as "an aggressive, intentional act carried out by a group or individual, using electronic forms of contact, repeatedly and over time against a victim who cannot easily defend him or herself". Another issue is generated by the bias of the already few publicly available datasets that are specialized on a specific type of aggression (e.g. sexism, racism, homophobia) or include data from only one social media platform. Finally, selecting the best combination of features that would work well on the training dataset as well as on future data is not an easy task due to changes in language or topic drift.

[1] The Bitdefender NU TASTA URA campaign site: nutastaura.bitdefender.ro, last accessed 25 July 2020.

RELATED WORK

The low number of publicly available datasets annotated for cyberbullying as well as a lack of variety among them, most of them targeting a single social media platform and containing out of context posts, make cyberbullying detection a complex problem to solve. That is the reason research mostly focuses on finding solutions for a specific social media platform or type of cyberbullying.

One of the more recent solutions regarding detection of cyberbullying on Twitter proposed by Chatzakou et al. [2] is based on the selection of features with a high discriminatory factor for more accurate classification. The dataset used for classification was collected over a period of 3 months and the goal was to label the users, rather than tweets. Therefore, a set of tweets posted by the same user was given to the annotator, that had to put the user into one of the categories: aggressive user, bullying user, spammer, or normal. This was done in order to consider the repeatability characteristic of cyberbullying, an accurate classification of bullying being hard to make without an ongoing communication between the bully and the victim.

Another study of Twitter posts was conducted by Al-garadi et al. [3]. A series of different binary classification models (Support Vector Machines, Random Forests, and Logistic Regression) and combination of features were tested to find the best classifier. Some of the features selected for training were number of followers, number of posts, number of tags, gender, and age of user. A ranking of those models and features shows that the most relevant features for classification were the presence of vulgarities and the gender of the user. We introduced such features in our solution as well.

Van Hee et al. [4] aimed to collect and use a dataset from Ask.fm, a social media platform based on question-answer type of posts, like Formspring from which one of our datasets is collected. A particularity of this study is that the manual annotators were guided to label the posts into a larger variety of categories that may indicate some form of cyberbullying, namely threats, insults, sexually explicit messages, texts encouraging a bully. Moreover, they introduce 2 new possible roles in the cyberbullying act: instigator (person that encourages a bully) and vigilante (person that steps up for the victim).

The importance of selecting an efficient combination of training features was illustrated in a study published by Dadvar et al. [5] where a dataset comprising comments collected from Youtube videos is used for training and analysis. Features regarding the user were found to bring high information value to the predictions, therefore they introduced features like the age and sex of the poster, or even the way in which they behave online, details like the time of the posting or the number of followers a user has, being also considered to be important.

In relation to the previous work, we propose to use traditional natural language processing methods as they are faster, explainable and require less computation. We aim to eliminate the shortcomings generated by the bias of using only a social media platform by combining models trained on 2 different datasets. Moreover, we intend to select the best combination of manually crafted features. To test our model architecture, we also collected Twitter posts over a period that are fed into our models to predict their class. At the end, we test a sample of those predictions against the results of manually annotations for the same data.

METHOD

Datasets

For training our models we combined 3 publicly available datasets. The first two were used for training the cyberbullying detection model while the third one, collected from Facebook, was used for training an aggressivity classifier. The aggressivity classifier was then used to compute the aggressivity factor feature for each post in the first two datasets. As it can be seen in the table below the number of positive examples, namely cyberbullying instances, in the training datasets is much lower than the number of negative examples, especially in the case of the Formspring datasets. In order to try and minimize the issues that may arise from this, we tried duplicating some positive examples and insert them into the datasets as well as change the class weight parameter of the model.

Dataset source	No. of classes	No. of entries	% of positive examples
Cyberbullying datasets			
Twitter	3	2 751	40.93 %
Formspring	2	12 773	6.08 %
Aggressivity dataset			
Facebook	3	12 000	57.90 %

Table 1. Datasets summary

Dataset source	Twitter	Formspring
Vulgarities	13.04%	18.21%
Vulgarities out of bullying posts	17.05%	65.72%
Aggressivity	65.53%	55.21%
Aggressivity out of bullying posts	72.29%	73.96%

Table 2. Vulgarity and aggressivity in cyberbullying datasets

After training the aggressivity classifier on the Facebook dataset, we proceeded to analyze the datasets which were used for the cyberbullying classification. Two factors were considered important, namely the presence of vulgarities and whether a post has an aggressive tone. Therefore, we used the aggressivity classifier to predict the classes of these datasets and a list of popular vulgar words. We can see in Table 2 that vulgarity is not directly correlated with cyberbullying, only 17.05 % of the bullying examples from Twitter also containing vulgarities. However, in the last row of the table we can see that over 70 % of bullying examples from both datasets are also considered to be aggressive, aggressivity being often associated with bullying.

Training Features

Inspired from previous works, we manually crafted and selected training features that are relevant for cyberbullying detection. These features were further grouped into different categories in order to test which type of features brings the most information to the cyberbullying classification. Those categories, along with the features included in them, are further explained in this subsection.

Content Features

- **Presence and frequency of vulgar words.** While there is not a clear relationship between vulgar words and cyberbullying, some studies that analyzed different feature combinations discovered that vulgarity related features are a high discriminatory factor for cyberbullying detection [6]. Another study focused on the presence of vulgar words on Twitter [7] found that compared to other social media platforms, Twitter posts have more vulgar words. This is relevant for our study too, considering that the datasets we collected include posts from Twitter.
- **Presence of hyperlinks.** We chose this attribute after analyzing a sample of bullying posts where hyperlinks were frequent. Most of the times these were links to photos in the form of rude memes addressed to a person.
- **Frequency of capital letters.** In the absence of information given in verbal communication, like tone and emphasis, we use other indicators to suggest the way we intend our message to be read. The use of all capital words or sentences is meant to put emphasis on certain words or may suggest a raised tone of the post's author. In relation with cyberbullying, the use of capital letters may be associated with an aggressive tone or yelling.
- **Superlatives.** Like capital letters, superlatives may be used in textual communication to emphasize a certain message. For this feature, we considered the presence of words indicating superlatives (e.g. *very*, *the most*) as well as the presence of words from a list of superlatives we comprised.
- **Post length.** We introduced this feature in order to create a clearer separation between shorter posts that are vulgar or aggressive and longer posts that have a lower aggressivity value but have a big change of being considered cyberbullying through their content.

Subjectivity Features

- **Presence of second person pronouns.** The use of second person pronouns indicates that the content of the post is directly targeting a specific person. In combination with insults or an aggressive tone, this might indicate the presence of cyberbullying.
- **Presence of first-person pronouns.** By analyzing the results of some early versions of our models we observed that some falsely classified as bullying posts were written in first person. Most of those were self-denigration, meant to be light-hearted self-jokes.
- **Presence of mentions.** Similar to second person pronouns, mentions also indicate who the content of a post is targeted at.
- **Vulgarity-Pronoun/Vulgarity-Mention pairs.** This feature is used to indicate whether a vulgar word was addressed to a person. For a post, we verify if a vulgar word is at most 2 words apart from a mention or pronoun. The list of vulgar words was created by collecting banned words from different social media platforms and is available online[2].

Aggressivity Features

The overall aggressivity of a post can be a good indicator of the author's mood, indicating a possible feeling of frustration or fury. Moreover, several studies consider aggressiveness as a definitory factor of cyberbullying [2, 3], while there is still much debate regarding differences between aggressivity and cyberbullying and whether they represent the same thing. The value for this feature was computed by using the aggressivity classifier trained on the Facebook dataset.

General Content Features

Finally, we introduced features regarding the words present in a post. To compute the values for these features we considered both single appearances of words, as well as pairs of words. We considered the most frequent 10k such n-grams, resulting in 10k different numerical features.

Algorithms

For selecting the best classification algorithm for this problem, we conducted several tests with different combinations of training features and algorithms and recorded the metrics for them. Since for cyberbullying detection we would rather have a larger number of false positive examples, rather than false negatives, the most important metrics for us are the recall and precision. Out of Support Vector Machines (SVM), Random Forest, and Logistic Regression, SVM had the best score overall and therefore was chosen as our classifier. More, SVMs are used in almost all studies of cyberbullying detection either as the main algorithm or in comparison to others [4, 6].

[2] The entire code is available online in the repository: https://github.com/stefanx17/cyberbullying_detection

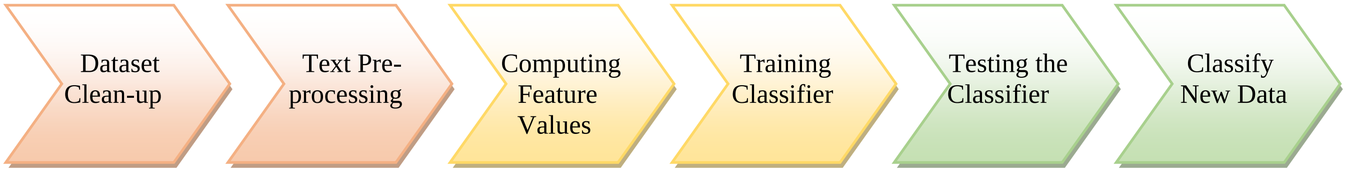


Figure 1. Cyberbullying classification steps

CLASSIFICATION PIPELINE DETAILS

Pre-processing and Feature Computing

Dataset Cleanup

This step was only applied for the dataset containing examples collected by us from Twitter. We considered that some examples may introduce unnecessary noise in the classification process and filtering them out would solve this issue. Therefore, we eliminated all retweets and duplicated entries from the training dataset.

Text Pre-processing

As shown in Figure 1, before computing the training features values, we cleaned-up the text by eliminating or replacing some elements in the text that do not bring any discriminatory information to the classification. We did this by using regular expressions applied to the text.

The modifications applied to the text are as follows:

- **Remove mentions and hyperlinks from the text.** Mentions represent references to other users and are marked in the text by the presence of the "@" symbol before the name of another user, while hyperlinks are marked by the presence of "http://" or "https://".
- **Restricting sequences of the same characters to a length of maximum 2.** Repeated consecutive appearances of the same character can either be typos or intended by the author to suggest the way the text is meant to be read. However, even in the latter case, there is no rule to how long that sequence can be, resulting in different length sequences that have the same meaning but are not considered to be the same when we compute features.
- **Elimination of unknown characters.** We chose to remove all non-ASCII characters.
- **Elimination of punctuation marks.** We remove all punctuation marks except the apostrophes or combinations of marks that may represent an emoticon.
- **Elimination of text sectioning rules.** This is applied only to the Formspring dataset where posts represent question-answer pairs, indicated by the presence of 'Q:' and 'A:'. We choose to eliminate those and consider the whole text of the post for training the model. We made this decision because splitting the text would make it hard to decide if bullying is present in the question or in the answer.

Hyperparameter	**Tested values**
Kernel type	{Linear; Poly; Rbf; Sigmoid}
C penalty	{0.001, 0.003, 0.01, ... ,100, 300, 1000}
Class weight	Default; Balanced

Table 3. SVM hyperparameters

Computing the Training Features

The only features that are computed before the text-processing are the ones regarding the presence of mentions and hyperlinks, as they are removed during the cleanup step. These are binary features, so their values are either 1, if these elements are present in the text, or 0 otherwise. The numerical features were computed by counting the number of appearances of certain words in the text of the post (the number of vulgar words, for example).

For the aggressivity feature, we used the classifier that predicts the aggressivity of a post on a scale from 0 to 1. We trained this classifier on the dataset specifically annotated for aggressivity and used it to classify all the posts in our training cyberbullying datasets.

Finally, for the features regarding the general content of a post we made use of modules designed for text feature extraction in the scikit-learn library [8]. We used the CountVectorizer module for determining the vocabulary of our dataset, composed of the most used unigrams and bigrams, and TfidfTransformer to get the final value for each feature.

Classifiers Architecture

To obtain the classifier used for detecting cyberbullying in a live collected dataset from Twitter, we needed to train and use several models, one for determining the aggressivity of a post and one for each cyberbullying dataset. As previously stated, we primarily selected SVM, for which we further did a grid search to select the best combination of hyperparameters. The hyperparameters we varied, as well as their values, can be seen in Table 3.

Aggressivity Classifier

For getting the value of the aggressivity feature we trained another model on a dataset annotated for aggressivity collected from Facebook. As features, we used TF-IDF using the most frequent 10k n-grams (n=1..3). As for the cyberbullying classifiers, the best results were obtained when using an SVM model.

Cyberbullying Classifier

In order to determine and select the best feature combinations we conducted several experiments by training the models with different features and comparing the metrics for each case. Before these experiments we conducted a grid search in the space of the hyperparameter values to determine the best values for the hyperparameters. The grid search was done by only using features described in the baseline configuration, namely TF-IDF features. All combinations have been tested, choosing variants that deliver the best results on each of the three machine learning algorithms used. After determining the values of the hyperparameters to be used when training our machine learning models, several tests were carried out, for each of the two classification models, training one at a time with different attributes. We will continue by presenting the attribute configurations chosen to be tested.

- **Baseline.** Only contains features consisting of unigrams and bigrams frequency (TF-IDF). We chose this as our baseline as these are the most common features for text classification.
- **Baseline + Content Features.** This feature configuration will show us how important is the content of a message (presence of certain words or phrases) for detecting cyberbullying.
- **Baseline + Content Features + Subjectivity Features.** In addition to the previous configuration we included subjectivity features, indicating who the content of a message was targeted at.
- **Baseline + Content Features + Subjectivity Features + Aggressivity Features.** This configuration contains all the features we computed for the datasets.
- **Baseline + Aggressivity Features.** We considered this configuration in order to find out how correlated is aggressivity with cyberbullying and to see if the aggressivity of a post is a strong enough feature to determine if it can be labelled as cyberbullying or not.
- **Content Features + Subjectivity Features + Aggressivity Features.** We want to see how well our manually crafted features do when we do not include TF-IDF features for training our models.

Collection and Classification of Twitter Data

For the last part of this research we intend to test our classifiers on a live dataset collected by us from Twitter. This data will be cleaned and pre-processed and then fed into our classifier to predict each post’s class. Lastly, we will compare the automatic detection against a sample of manually annotated data to see how well our classifier performs on real live tweets.

Collection of Twitter Datasets

For testing our classifier, we decided to collect new data from Twitter that we will annotate both automatically with our trained classifier and manually with 3 different human annotators. Therefore, we collected two datasets, one with random posts and one by searching for keywords that may indicate the presence of cyberbullying. The keywords were taken from a list that contains the most used words in cyberbullying posts from the datasets we used for training. To increase the possibility of a higher variety of topics, we collected posts every day for a period of 6 weeks in May-June 2019. For data collection, we use the API provided by Twitter and tweepy (https://www.tweepy.org/), a Python library that provides methods for post searching.

Because the datasets we use for training our models mostly contain English posts, we chose to only collect posts written in English. Therefore, the filters we introduce in our posts search call were the language of the post and a list of keywords for one of the datasets.

Automatic Classifier Architecture

For the classification of the newly collected data we decided to use two previously trained classifiers, one on the Formspring dataset and the other on the Twitter dataset (see Table 1). From both those classifiers we obtained a probability of the presence of cyberbullying in a tweet that we combined to obtain a final prediction. We did this by computing the median of those predictions, except for when at least one of the classifiers predicted the cyberbullying class with a very high confidence (more than 85%), in which case the post was automatically considered to be cyberbullying even if the median indicated otherwise. We consider this exception to be necessary as one of the classifiers might not be sensible to a certain type of cyberbullying, resulting in a lower cyberbullying score even if the post has some cyberbullying indicators.

Manual Annotation of Posts

From each dataset we selected a sample of 100 posts that had the highest cyberbullying score to be manually annotated. Therefore, in the first dataset collected based on the presence of words that may indicate cyberbullying all the selected posts were labelled as cyberbullying by the automatic classifier, while only 45% of posts in the random dataset were manually labelled as cyberbullying.

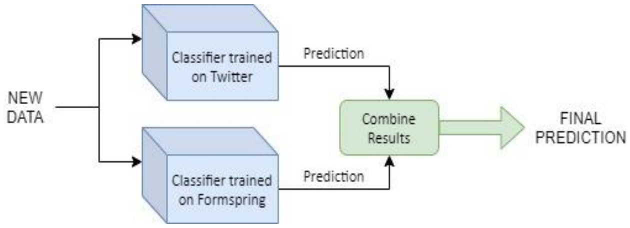

NEW DATA
Classifier trained on Twitter
Classifier trained on Formspring
Prediction
Prediction
Combine Results
FINAL PREDICTION

Figure 2. Cyberbullying classifier architecture

Bullying
The presence of cyberbullying is obvious and one or more of the indicators are present: • Humiliation of another person • Vulgar words • Indicators that suggest the repeatability of cyberbullying • Aggressive tone used by the author of the post • Other indicators considered to be relevant by the manual annotators
Maybe Bullying
The presence of cyberbullying is not obvious, but some indicators of cyberbullying are present in the text of the post. Some relevant indicators are: • One or more indicators presented previously in the definition of cyberbullying • General references that intend to humiliate/harm a category of people (by race, sexuality), but are not directed at a specific person (e.g. “All gays must be killed”) • The possibility to identify a context in which that post might be considered cyberbullying (e.g. „Just end it already” might be an allusion to suicide in some cases)
Not Bullying
It is obvious that cyberbullying is not present, as none of the previous indicators are satisfied.

Table 4. Cyberbullying classification guidelines

After selecting the posts, we determined 3 classes in which the manual annotators can sort the posts they were given. For the manual annotation, 3 people were asked to determine the class of each post based on the text of the post as well as on the guidelines described in Table 4.

Finally, after the manual annotation, we analyzed the results and computed metrics of the given answers. One of the metrics we computed was the inter-rater agreement rate, used to determine the similarity of the answers from different annotators. The second one refers to the frequency of cyberbullying according to our classifiers' opinions. To determine this percentage, we split our samples in groups of 20 posts and analyzed them individually.

To compute the acceptance rate between the classifiers, each posts received a score:

- 3: If all the annotators chose the same label for the post
- 1: If at least two of the classifiers chose the same label
- 0: If all the classifiers chose a different label for the post

Finally, we combined these scores and obtained an overall acceptance rate by using the following formula:

$$Acceptance\,rate = \frac{\sum_{p \in Posts} score_p}{||Posts|| \times MAX\_score}$$

RESULTS

Cyberbullying Classification Results

The configurations selected for testing our detection algorithms can be seen below and were further explained in the previous section. For each new combination, we maintained the previous features and added a new category until we reached a configuration where we use all the features selected by us (D). Then, we also test how aggressivity features alone influence the results and we also test a feature combination where we don't use TF-IDF.

A. Baseline (TF-IDF)
B. Baseline + Content Features (CF)
C. Baseline + CF + Subjectivity Features (SF)
D. Baseline + CF + SF + Aggressivity Features
E. Baseline + Aggressivity Features
F. CF + SF + Aggressivity Features

The performance metrics are precision, recall, f1-score, and accuracy. In the case of cyberbullying, the most relevant metric is the recall, as we want the percentage of undetected cyberbullying posts to be as low as possible. The f1-score is also relevant as we want to maintain a balance between precision and recall, too many posts wrongly labelled as cyberbullying not being good either.

The results of using our classifier architecture on the two different datasets can be observed in Table 5. On a first look, we can clearly see that better results were obtained for the Twitter dataset, the Formspring dataset having a much lower precision, thus ignoring many of the cyberbullying posts. Some of the reasons for this difference could be concerning the different platforms these datasets were collected from, as well as different periods of times, and different guidelines for manual annotation. Analyzing the different feature combinations, we can see a small improvement in adding subjectivity and content features, especially in the case of the Formspring dataset.

Cfig Metric	A	B	C	D	E	F
Twitter dataset						
Precision	63.5	63.2	**64.9**	64.4	63.4	49.5
Recall	**82.4**	79.6	80.5	80.5	78.7	50.9
F1	71.7	70.4	**71.9**	71.6	70.2	50.2
Accuracy	74.6	73.9	**75.3**	75.0	73.9	60.5
Formspring dataset						
Precision	34.0	35.4	36.1	**37.1**	34.3	36.0
Recall	63.0	60.2	64.3	63.0	60.2	**65.7**
F1	44.2	44.6	46.3	**46.7**	43.7	46.6
Accuracy	90.9	91.4	91.4	**91.7**	91.1	91.3

Table 5. Classification test results

	Bullying Sample May-June 2019	Random Sample May-June 2019
Sample size	6783	3144
Bullying %	6.04 %	1.46 %
Aggressivity %	48.68 %	55.69 %

Table 6. Results of automatic classification

	Bullying Sample May-June 2019	Random Sample May-June 2019
Bullying %	53.00 %	36.00 %
Inter-rater Reliability	67.33 %	68.67 %

Table 7. Results of manual classification

Twitter Sample Classification Results

Automatic Classification Results

As it is illustrated in Table 6, the dataset collected by searching for words that appear often in cyberbullying posts has a higher percentage of posts classified as cyberbullying, almost 9 times more posts than in the case of the randomly collected samples from Twitter. In the case of aggressivity, both datasets have a high presence of it in their entries, about half of the posts being labeled as aggressive. Even though this percentage is troubling, the fact that is way different than the percentage of cyberbullying further shows that these two phenomena should not be confused with each other, each having their particularities.

Human Classifiers Results

To obtain the following statistics, we selected a sample of 100 posts from each dataset by the probability of cyberbullying assigned to each of them by the automatic classifier. Before presenting them to the human classifiers we randomized their order and hid their cyberbullying score. Then, the human classifiers were asked to assign each post to one of the three possible classes: Bullying, Maybe Bullying, Not Bullying by following the guidelines presented in Table 4. As we consider important in cyberbullying detection to consider any indication that this phenomenon appears in a post, for the statistics below we label the post as cyberbullying if at least 2 out of the 3 annotators classified it as Bullying or Maybe Bullying.

As with the automatic detection, the random sample has a lower presence of cyberbullying, this is partly due to the fact that more than half of the posts were not classified as cyberbullying by the automatic detector either.

Also, we can observe in Table 7 that the inter-rater reliability score is quite high, being higher than 65% for both datasets. This indicates that people have an acceptable rate of agreement when it comes to identifying cyberbullying. The reason why we don't have a higher rate of agreement can be the fact that this is a hard phenomenon to identify in small posts taken out of context, even when it comes to classifiers presumed to have a clear understanding of cyberbullying and its indicators.

Finally, considering the relatively low percentage of bullying found in the randomly collected sample (1.46 %), we think that a solution based on automatic detection, paired with human classifiers in the form of moderators on different social media platforms could help reduce the negative impact of cyberbullying on the Internet. A simple and straightforward solution would be having a classifier app triggered by common indicators of cyberbullying (similar to how we collected our bullying sample), then a moderator should be notified of any possible presence of cyberbullying behavior in order to investigate and take a final decision regarding the suspected post.

CONCLUSIONS

Cyberbullying has become an increasingly larger problem in recent years, affecting the mental health and safety of people, especially when it comes to children and teenagers. Therefore, solutions to best handle this situation and try to solve are of high interest for many organizations. Thus it has become a topic of interest for machine learning research meant to detect cyberbullying as accurately as possible.

After conducting several experiments with different training features combinations, we found out that the features introduced by us improve the classifications, in all cases the metrics being better than the baseline configuration. However, these must be used in combination with general content features (TF-IDF) in order to provide good results.

Our next goal was to engineer a model capable of accurately classifying new data and not be biased towards a certain social media platform. We tried to eliminate this bias by combining classifiers trained on 3 different social media platforms (Twitter, Formspring, and Facebook) in order to get a final prediction. To test out the proposed method, we collected two new datasets from Twitter (one random and one based on cyberbullying indicators) that we automatically classified with our model. We compared the results of automatic classification against a sample of manually annotated posts and we discovered that more than 50% of the posts detected as cyberbullying were also labeled as cyberbullying by human annotators. We consider this score to be very good, as automatic detection can be doubled by a human moderator that can make a final decision and decide whether to take action or not.

REFERENCES

1. Smith, P. K., del Barrio, C., & Tokunaga, R. S. (2012). Definitions of Bullying and Cyberbullying: How Useful Are the Terms?. In Principles of Cyberbullying Research (pp. 54-68).
2. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017). Mean birds: Detecting aggression and bullying on twitter. In Proc. ACM (2017) on web science conference (pp. 13-22).
3. Al-garadi, M. A., Varathan, K. D., & Ravana, S. D. (2016). Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. Computers in Human Behavior, 63, 433-443.
4. Van Hee, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, B., ... & Hoste, V. (2018). Automatic detection of cyberbullying in social media text. PloS one, 13(10).
5. Dadvar, M., Trieschnigg, D., Ordelman, R., & de Jong, F. (2013). Improving cyberbullying detection with user context. In European Conference on Information Retrieval (pp. 693-696).
6. Dadvar, M., Jong, F. D., Ordelman, R., & Trieschnigg, D. (2012). Improved cyberbullying detection using gender information. In Proc. Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012).
7. Wang, W., Chen, L., Thirunarayan, K., & Sheth, A. P. (2014). Cursing in english on twitter. In Proc. 17th ACM conference on Computer supported cooperative work & social computing (pp. 415-425).
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 2825-2830.

Stroke Detector - An Application that applies the F.A.S.T. Test to Identify Stroke

Mihnea-Ioan Rezmeriţă, Irina-Elena Cercel, Adrian Iftene

Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iaşi, Romania

{mihnea.rezmerita, irina.cercel, adiftene}@info.uaic.ro

DOI: 10.37789/rochi.2020.1.1.7

ABSTRACT

In Romania, stroke is the second leading cause of death and disability (Neagu, 2019). The speed of reaction in the case of a stroke can make the difference between life and death, between a healthy patient and a patient who remains disabled for life. In this context, we wanted to create a mobile application in Romanian that will help users to do a F.A.S.T. test very quickly. The test identifies possible problems at the face, arms and speech levels. This article shows how we made this application using a client-server architecture and what users think about it.

Author Keywords

F.A.S.T., Client-server architecture, Stroke, Face detection, Speech recognition.

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces. H.3.2. Information Storage and Retrieval: Information Storage.

General Terms

Human Factors; Design.

INTRODUCTION

We now live in a world where technological progress makes our lives easier and better. We live in an era where almost anything is possible, an era in which there are no restrictions of any kind, and software and hardware development amazes us from day to day. Mankind has progressed extremely fast in the last 20 years, and fields such as medicine have evolved a lot.

Medicine is more than vital to us because it directly contributes to our quality of life. For this reason, medicine is one of the sciences most often approached by researchers in all fields, in order to improve existing tools. Over the past 20 years, researchers have been able to successfully solve some of the most important problems and mysteries in the medical field, eradicating various diseases and reducing the risks of certain diseases that were once considered lethal. A crucial role in the development of medical systems was played by the field of informatics. Computer science, like other sciences, has benefited from an astonishing growth, which is why the fields that relied on it (including medicine) have developed even more. So computer science has helped on the one hand to finalize tools to help medical subfields such as Medical Imaging, Epileptology, Oncology or on the other hand has helped to develop tools capable of influencing the ability of doctors in various other subfields.

This paper addresses a topical issue in the medical field, namely the detection of stroke. Even today, stroke is an extremely difficult problem for doctors because time is a decisive factor for the long-term evolution of the patient. In the case of stroke, the recognition of symptoms at an early stage, as early as possible determines whether or not the patient will be left with certain long-term sequelae (Weber, 1995). The current solutions address cerebral stroke detection and monitoring using cloud services (García et al., 2019).

WHAT IS STROKE?

Stroke is a very serious condition that endangers the proper functioning of the body and in many situations can lead to a disability or even lead to death[1]. It is found in all age groups, although mostly in the elderly. It is important to note that a crucial role in the occurrence of stroke is played by the medical history of the victim and also the medical history of first-degree relatives (Clarke and Forster, 2015).

General description

Stroke occurs when blood supply to a portion of the brain is interrupted (Marin, 2019). Discontinuation of blood supply for a long time causes the death of nerve cells in the affected section of the brain, leading to disabilities. In

[1] https://www.nia.nih.gov/health/stroke

many countries stroke[2] is considered to be the number one cause of disability. There are several types of stroke[3] (see Figure 1) (Sorenson et al., 2019):

- **Ischemic stroke -** accounts for approximately 87% of all stroke cases and occurs when an artery that irrigates nerve tissue is blocked.
- **Hemorrhagic stroke** - occurs when certain blood vessels rupture and cause internal bleeding. This causes a lot of pressure on the brain and causes a massive loss of blood in the surrounding areas.
- **Transient stroke**[4] - is also known as mini-attack and is caused by a temporary blockage of blood vessels leading to the brain. It should not be ignored, as it can be a potential symptom of a future ischemic stroke.

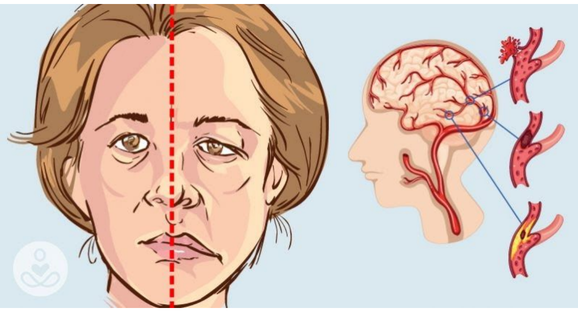


Figure 1: Stroke types[5]

Stroke can occur in a variety of forms and can vary from patient to patient. However, each patient will in turn experience one or more symptoms as well[6]:

- *Confusion* - generalized and cognitive problems;
- *Speech problems* - a visible difficulty in pronouncing words correctly;
- *Weakness* - some incapacity partial (or complete) to move a particular member;
- *Facial asymmetry* - it often happens that the mouth is dropped to one side;
- *Sudden and severe headache*.

Risk factors

Stroke is one of the leading causes of disability. However, it can be prevented by avoiding risk factors[7]:

- *High blood pressure*;
- *Smoking* - which is also a triggering factor in the event of other diseases;
- *Diabetes* - many people with diabetes have high blood pressure, are overweight or have high cholesterol;
- *Sedentary lifestyle* - which causes other problems;
- *Obesity* - which can still be combated with the help of a healthy lifestyle;
- *Heart disease* - of course, a medical history specific to heart disease is an important factor in the occurrence of stroke.

Forecasts and the importance of timely action

It is very important to act knowingly from the first sign of a stroke (Shugalo, 2019). We know that a stroke leads to the death of nerve tissues, so any wasted second is to the detriment of the patient.

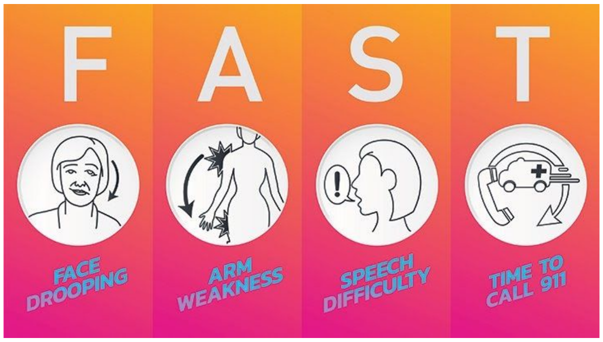


Figure 2: FAST Test[8]

Some of those who suffer a stroke are left with sequelae, consequences, such as various paralysis or cognitive and speech problems (Brown, 2002). In order to be able to act in the shortest possible time, the test F.A.S.T.[9] was developed (see Figure 2), an abbreviation that aims to incorporate all the main symptoms:

- F - Face asymmetry;
- A - Arm weakness;
- S - Speech problems;
- T - Time to call the ambulance.

This paper presents the implementation of an application in Romanian that is able to perform an analysis similar to that performed by the F.A.S.T. to be able to act in time.

[2] https://www.stroke.org/en/about-stroke

[3] https://www.laboratorpraxis.ro/bine-de-stiut/tipuri-de-accident-vascular-cerebral

[4] https://www.stroke.org/en/about-stroke/types-of-stroke/tia-transient-ischemic-attack

[5] https://www.dornamedical.ro/images/articole_media/art_neuro_iunie_2019/POZA%20AVC%206.jpg

[6] https://www.medlife.ro/glosar-medical/afectiuni-medicale/accident-vascular-cerebral-avc-cauze-simptome-tratament

[7] https://www.stroke.org/en/about-stroke/stroke-risk-factors

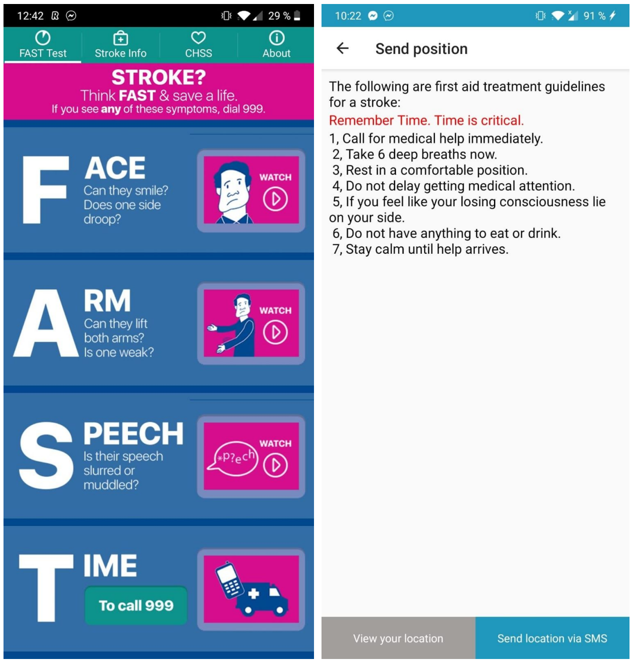
[8] https://image.shutterstock.com/image-vector/stroke-warning-signs-symptoms-icon-260nw-1008179179.jpg

[9] https://www.stroke.org/en/about-stroke/stroke-symptoms

EXISTING SOLUTIONS

Fast Test

This application[10] was developed by CHHS[11], an organization that provides services to those affected by stroke and to raise awareness in Scotland about various general aspects of stroke. As seen in Figure 3 (in left), the colors are very appropriate, combining a bright pink with a cooler color.



STROKE?
Think **FAST** & save a life.
If you see **any** of these symptoms, dial 999.

F ACE
Can they smile? Does one side droop?

A RM
Can they lift both arms? Is one weak?

S PEECH
Is their speech slurred or muddled?

T IME

Send position

The following are first aid treatment guidelines for a stroke:
Remember Time. Time is critical.
1, Call for medical help immediately.
2, Take 6 deep breaths now.
3, Rest in a comfortable position.
4, Do not delay getting medical attention.
5, If you feel like your losing consciousness lie on your side.
6, Do not have anything to eat or drink.
7, Stay calm until help arrives.

Figure 3: FAST Test App (in left) and S@S App (in right)

The application aims to present the details related to each component of the F.A.S.T. to those who install it. In addition to descriptions and suggestive images, the application also has short videos for presentation.

Although the application looks good and behaves just as well, it has one major drawback: it does not test the appearance of the stroke, but only presents it. We consider the presence of a component to perform an F.A.S.T. would have brought a very useful functionality to it.

S@S

Another application is SPOT@STROKE[12]. It was developed by user Kieran O'Callaghan to help people who are suspected of having a stroke. This project has an interesting approach based on a checklist that presents the main symptoms of stroke. The app also has the option to take a few photos relevant to the F.A.S.T. test, but does not process or use them to identify the stroke. A very good idea of the application is the presence of a close contact list. This list is very useful when we need to contact an acquaintance of someone who has such a stroke.

Another interesting component is the Send Position option that allows users to send the coordinates to which they are to a specific contact in the contact list (see Figure 3 in right). As with the previous application, there are some useful features, but it also lacks the F.A.S.T. attesting to the presence or absence of a stroke.

PROPOSED SOLUTION

The application we made is built on a server capable of detecting the most important features of the F.A.S.T test: facial asymmetry, arms weakness and speech problems.

System Architecture

The application uses a client-server architecture (similar to architecture from (Siriţeanu and Iftene, 2013)). The client sends requests to the server, which it receives, processes them, and finally sends the response back (see Figure 4).

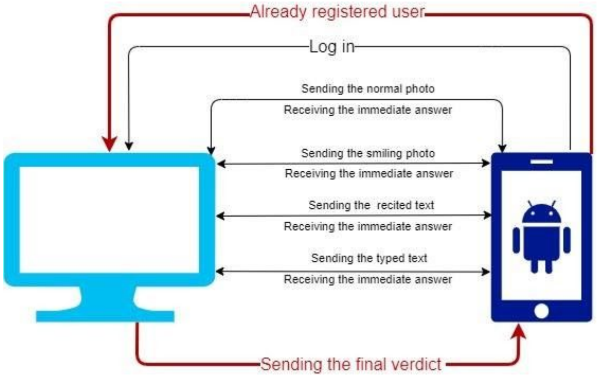

Already registered user
Log in
Sending the normal photo
Receiving the immediate answer
Sending the smiling photo
Receiving the immediate answer
Sending the recited text
Receiving the immediate answer
Sending the typed text
Receiving the immediate answer
Sending the final verdict

Figure 4: System Architecture

How the application works

In order to use the application, a user must create an account in advance. When he creates an account, he will go through several steps, such as taking a normal picture of him, taking an image in which he smiles, reading a text and typing a text. This information will form the user's profile and will be used as a reference when he wants to take an F.A.S.T. The F.A.S.T. involves following steps similar to those performed when creating the account:

- Take a normal picture of the user at the mobile device level and send it to the server. Its processing is done on the server, and it is

[10] https://play.google.com/store/apps/details?id=uk.org.stroke.fasttest
[11] https://www.chss.org.uk/
[12] https://play.google.com/store/apps/details?id=com.whereiskieran.android.spotamaster

compared with the profile image in order to detect the facial asymmetry specific to the stroke;

- Take a picture of the user smiling at the mobile device and send it to the server. On the server it is processed and compared with the profile image in which the user smiles in order to detect muscle weakness in the face;
- Retrieve user-readable text as an audio file at the mobile device level and send it to the server. The server compares the text extracted from the audio file with the text that the user had to read in order to detect speech problems;
- Retrieve a text typed by the user after a text received by him at the mobile device level and send it to the server. The server compares the initial text with the text typed by the user in order to identify the weakness of the arms.

At the end of the test the user will receive a warning in case he is suspected of a stroke and the necessary advice to help him manage the situation. If clear signs of a stroke are detected in one of the above steps, the test stops and skips to the last warning step and provides advice on how to handle the situation. At the client level there is also an information component, where the general symptoms of the stroke are presented, through a slide show.

Each component will be analyzed in more detail in the following chapters.

Server Component

The server was created using Python and Flask[13]. The server consists of several endpoints, all secure, so that they can only be accessed by authenticated entities. The server uses a database for user authentication, an authentication system, a local photo storage system, and various classes needed to parse the information and determine the final decision.

The server consists of several packages, files, modules and classes, each with a well-established role in stroke detection. Its purpose is to deal with requests submitted by users:

- login - Connecting and obtaining the necessary tokens, supports only POST type requests;
- register - User registration, supports only POST type requests;
- check_symmetry_normal_img - determination of facial features with the help; picture sent by the user, only supports POST requests;
- check_smiley_corners - determining the degree of muscle weakness in the face with the help of the picture sent by the user, supports only POST requests;
- get_text - returns a text to be typed or recited, supports only GET requests;
- parse_voice - determining the existence of speech problems using the text sent by the user, supports only POST requests;
- send_texting_test - determining muscle problems using text sent by the user, supports only POST requests;
- send_final_result - returns a final result, only supports GET requests.

Applications specific to the FAST test will return partial scores, which will be used to calculate the final score and make the final decision.

Android Client

The client is an application that runs on an Android device. The Android client offers an attractive and easy to use interface, including the F.A.S.T. and another special component for presenting symptoms. The user is presented at the beginning, the home page, which exposes the 2 options (Figure 5 in left):

- Conectare (Login) - in which the client is authenticated;
- Înregistrare (Registration) - through which he can create a new account.

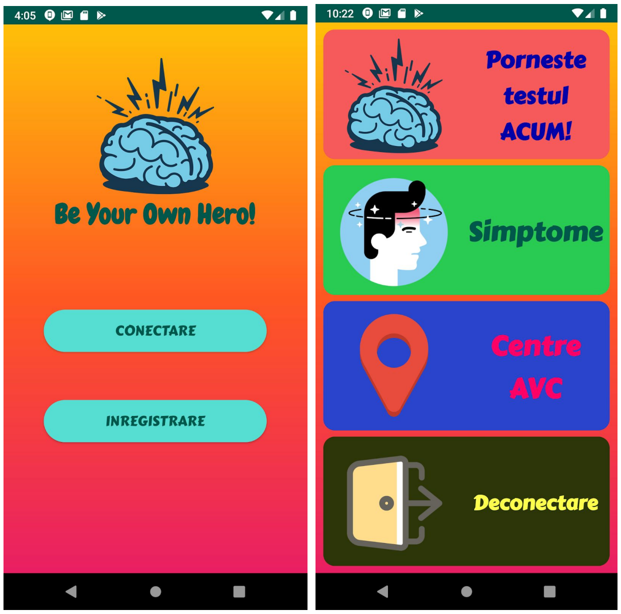


Figure 5: Home screen (in left) and Main menu (in right)

After connecting, the user is presented with the main menu, which allows several options, as shown in Figure 5 in right:

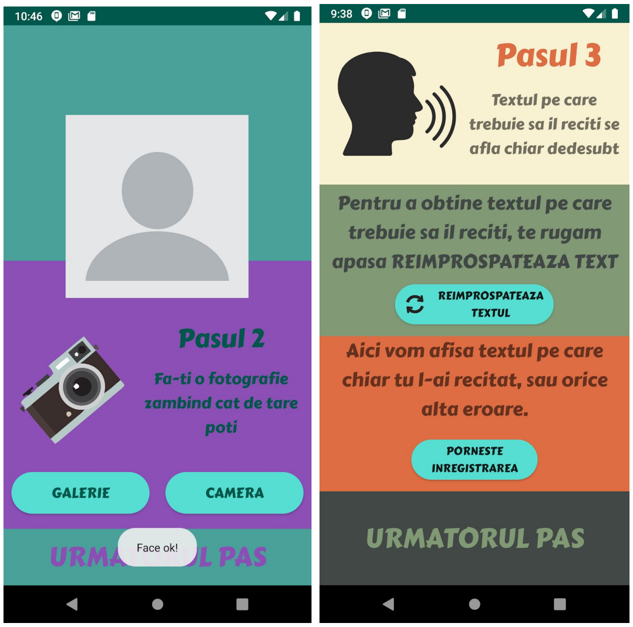
- F.A.S.T. test;

[13] https://flask.palletsprojects.com/en/1.1.x/

- Symptoms;
- Map of stroke centers (currently under construction);
- Disconnect.

FAST Test

The first step of the test is to send a normal picture of the test subject. He can choose from two options: either take a picture on the spot, or choose one from the gallery. The user is required to choose a picture in order to proceed to the next stage of the test. Once he sends the normal picture, in step two he is asked to send a picture in which he smiles (see Figure 6 in left).



Pasul 2

Fa-ti o fotografie zambind cat de tare poti

Pasul 3

Textul pe care trebuie sa il reciti se afla chiar dedesubt

Pentru a obtine textul pe care trebuie sa il reciti, te rugam apasa REIMPROSPATEAZA TEXT

Aici vom afisa textul pe care chiar tu l-ai recitat, sau orice alta eroare.

Figure 6: Step 2 (in left) and Step 3 (in right)

After these first two steps, the speech testing component follows (see Figure 6 in right). Here, the user's transition to the next stage is conditioned by 2 elements: obtaining the text to be recited and reciting it.

The user will initially get a text by pressing the *Refresh text* button. The required request will be called and the information obtained will be displayed. The second part of this step is the actual recitation of the obtained text, after pressing the *Start Recording* button, so that the device is aware that it needs to listen. Subsequently, the device will parse the recording in a very short time (approx. 2-3 seconds), and will display the resulting text. It is important to note that some audio recording parsing errors may occur, but they will be displayed so that the user understands the reason for the error.

The last step of the test is to detect the weakness of the arms, through a typing test. This test is similar to the previous one, as it consists of 2 components: obtaining the text, and typing it.

After completing this step, the user will receive the final verdict so that he knows if he is in danger or not. Depending on this we will receive some tips and suggestions, and the option to call a friend or even call 112.

Permissions

During the development of the client application it was important to consider the permissions, as the application needs to access various features specific to the Android platform. The application needs the user's consent for features such as:

- accessing the location;
- internet access;
- sound recording;
- access to the room.

Without these permissions, the F.A.S.T. cannot be achieved. Therefore, whenever needed, the user will be asked if it allows the application to use certain specifications. In case of a refusal, the functionality of the application is reduced, becoming almost useless. So, only if the user agrees to offer these permissions, the application can follow the course described above.

F.A.S.T. TEST

This chapter is responsible for explaining how we get facial features or how to achieve speech-to-text transformation..

Detection of facial features

One of the main components of the application is the detection of facial features. Detection of facial features is a problem often addressed by researchers, for which there are various viable solutions today (Asteriadis et al., 2011), (Barra et al., 2018), (Omer et al., 2019).

In our case, we used already predefined models and trained inside the dlib library[14] (Rosebrock, 2017). These models have a very good accuracy and are very effective in solving this problem[15] [16].

The chosen model was trained on the iBUG 300-W dataset [17], and determines 68 important points in front of a man, points that represent the following components (Barra et al., 2018):

[14] http://dlib.net/

[15] https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[16] https://www.geeksforgeeks.org/opencv-facial-landmarks-and-face-detection-using-dlib-and-opencv/

[17] https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

- lip;
- eyebrows;
- eyes;
- jaw;
- nose.

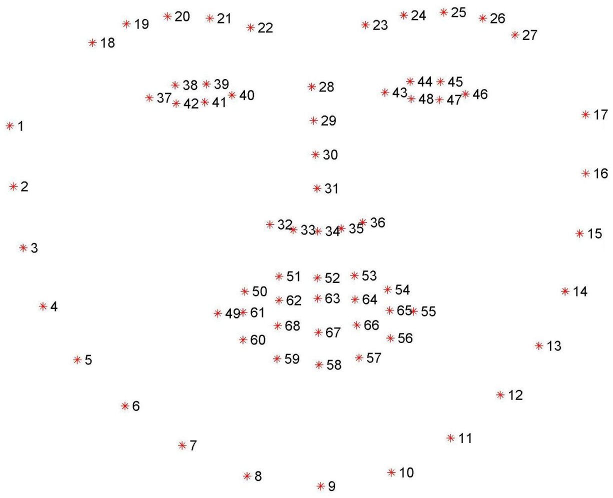


Figure 7: The 68 coordinates (Barra et al., 2018)

Also, the model used is a special model, already trained, which detects only the faces in the image, without a background. For this, the initial image is processed by the face detector, so as to restrict the area of interest in the picture. Having marked this area of interest in the picture, only it will be sent to the model that detects the specific elements of the faces. The algorithm after dialing will return a vector of 68 elements specific to the identified face (see Figure 8).

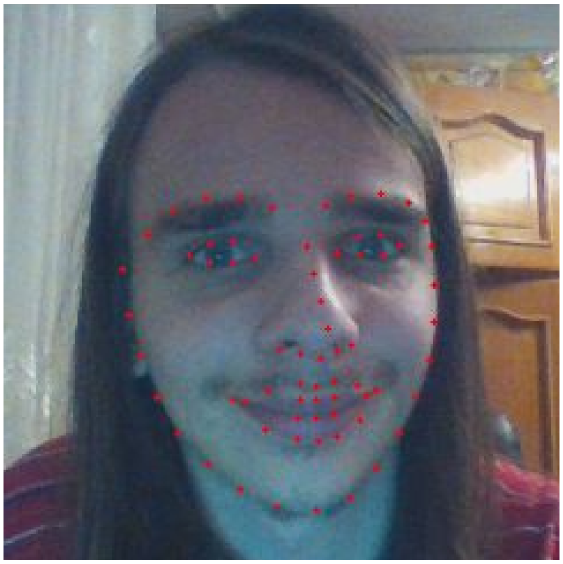


Figure 8: The 68 coordinates on an example

All these coordinates are part of a certain feature that we are looking for, for example, the mouth - the coordinates in the range [0-12], etc. Thus, the application determines the coordinates of these features.

Detection of face asymmetry

Knowing the above coordinates, the application is able to detect facial asymmetry. For the picture taken initially when setting up the user account and the current picture taken when performing the test, the positions of the eyes, the position of the mouth, the position of the corners of the mouth will be compared.

For the identified differences, an average of the coordinates on the left side of the mouth and an average of the coordinates on the right side of the mouth are calculated. Then the difference between the two parts is calculated. Similarly we do the same calculations for the eyes. These differences will give us a local score in the mouth, eyes, etc. Then with these local scores a global score will be calculated, which will help us in establishing the final verdict.

The most important local score is that of mouth asymmetry, which is associated with a common signal in the identification of stroke. Asymmetry of eyes, eyebrows, etc. represents secondary signals with less importance in the calculation of the final score.

Thus, if the mouth has a high degree of asymmetry, the application will decide that we have identified the stroke. If the asymmetry is average, the asymmetry of the eyes and other components will also be taken into account to calculate the final score. Also, a strong asymmetry in the eyes will also be sufficient for the detection of stroke, because the asymmetry of the eyes is also an important signal of the existence of stroke.

After several experiments and tests with the data we collected, we established the thresholds for the partial scores and for the final score, which told us that we have or have not identified a stroke attack in a patient.

Detection of muscle weakness in the face

Detecting muscle weakness in the face is not part of the F.A.S.T. test standard, but it is an approach that we have experienced in our tests and that could give good results in the future. To detect muscle weakness in the face, we check if the user is able to smile or not. For this, we compared the distances from the levels of the corners of the mouth. If the distance is above a certain threshold determined by us experimentally, then it means that the user does not have a problem. However, we must take into account the fact that some people smile differently, that is, they do not spread the corners of the mouth, but rather separate the lips so that the teeth can be seen. That's why we took this fact into account, calculating the distance between the upper and lower lip. If this distance is also too small, the user may have a problem.

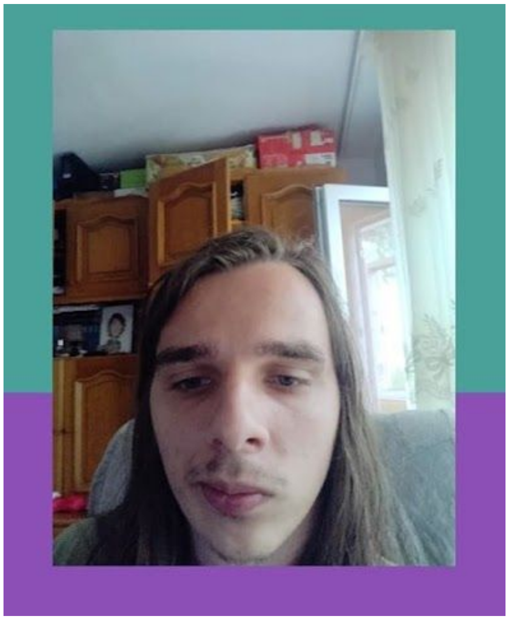


Figure 9: Smile attempt

To calculate these distances we also used the 68 coordinates from (Barra et al., 2018) and the images initially saved when the user created his account and the current image in which he smiles. Figure 9 is an attempt at a half-mouthed smile, which after the evaluation will notify the user that he is in danger.

Detection of speech problems

Another very important component of the test is the way in which speech problems are determined, similar to (Ţucă and Iftene, 2017). The user receives, at the client level, a text, which will have to be read by him. While the user is reading the received text, the client on the mobile phone will process the audio signal and turn it into text, using the component SpeechRecognizer[18] from Google. The resulting text will be sent to the server, along with the text received for recitation. The server will calculate the number of words that were mispronounced in relation to the text that was originally read.

Then, the number of mispronounced words will be used initially to calculate a partial score, and then to calculate a final score. Similar to the previous components, it can be decided based on the partial score whether we detected a stroke or not. The score threshold for determining whether or not we have a stroke was also identified after several tests and experiments.

Detection of muscle weakness in the arms

Muscle weakness in the hands is detected in a manner

similar to that used in the component presented in the previous point. The user requests a text, which he will have to type, after which it will be sent together with the initial text to the server. If the user has problems with the hands, then he should not be able to type it.

At the server level, we used the Levenshtein distance (Levenshtein, 1966) to determine the differences between the two texts. These differences help us calculate a partial score for this component and then a final score.

After several experiments we determined that if the user makes at least half of the number of letters he had to type, then he is suspected of having a stroke.

The final decision

The final decision is the decision we make either based on a partial score that is above a certain threshold, or based on the final score, when it also has a value above a global threshold. We obtained all these thresholds experimentally after several tests and experiments that we did with several users..

USABILITY TESTING

Age categories

The application was tested by 22 people, in a direct way interacting with it but also indirectly by watching a video showing all the capabilities of the application. The interviewees are from different age groups as shown in the chart below.

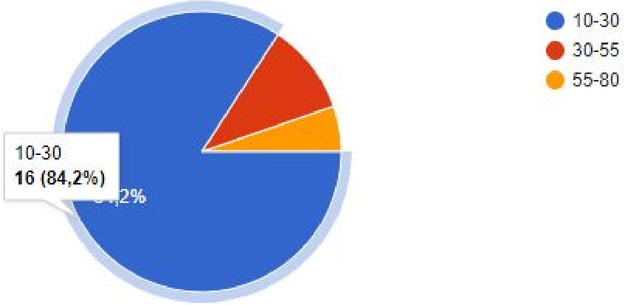


Figure 10: Age categories

The high percentage of people in the age category 10-30 is noticeable, but the other 2 categories should not be ignored either: 30-55 and 55-80 each with a percentage of 10.5% and 5.3% respectively. Everyone thought that the

[18] https://developer.android.com/reference/kotlin/android/speech/SpeechRecognizer

application was useful and that it could have a real impact on people's lives.

How intuitive is the application?

Many of the interviewees found the application intuitive and easy to use, but there were also people who reported that the application could be improved in terms of UI component and mode of operation, as reported in the chart below:

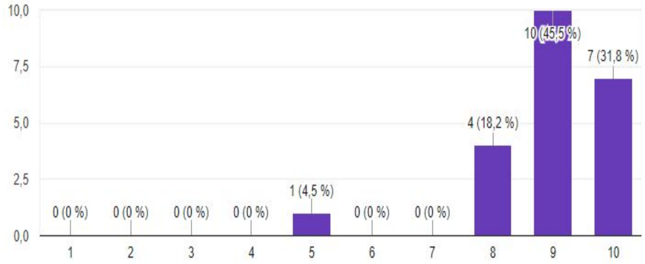


Figure 11: How intuitive is the application?

Regarding the waiting times, the opinions were mostly positive. Most felt that they did not have to wait long and others, fewer, considered that it took too long, especially when the Internet connection was not very good..

How do you like the design of the application?

From an aesthetic point of view, the design of the application was appreciated by users, they were pleasantly surprised. However, we believe that following the opinions gathered, it can be improved.

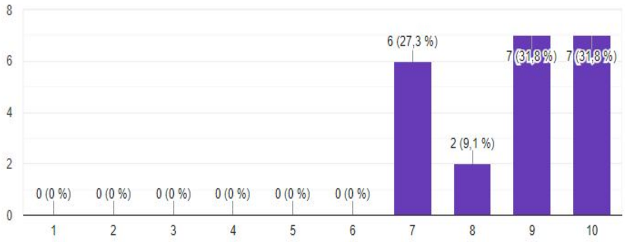


Figure 12: How do you like the design of the application?

All users considered that the application is of vital importance, and that they would use such an application if they were suspected of having a stroke. Among the most important improvements that users consider necessary are: the introduction of a machine learning component and the introduction of a voice narrator to guide users. Other improvements that could be made were the elimination of the recording component, the optimization of the symptom component and the introduction of voice commands.

Important future improvements

Also, many of the users interviewed preferred to express their opinion in a free way on things that could be improved. Among the things they listed are the following:

- Messages like "Call 112" or "We're sending the information to the server!" they should be placed somewhere at the top of the page so that the user can see the information more easily;
- Another suggestion would be to introduce more types of tests, the application contains the FAST test, but to make it more extensive, it could also contain other types of more complicated tests that test certain aspects in more detail, over a longer period of time, in order to be able to prevent a possible attack even before it occurs, to see the evolution in time of the user;
- If the elderly want to use the application for example, they may encounter problems such as: they do not understand the written instructions, the text is not large enough, they have difficulties in taking the picture. The idea with the vocal narrator is very good, but we consider that we have to repeat the commands when we do not have user interaction for a long time. We think a "Quick Test" option would be good (a quick test without the need for registration). Some font size settings or a preference for written / audio instructions would be helpful.

General opinion of the users interviewed

In general, users were satisfied with the initiative brought by the application, its purpose, how it works, how it behaves and the solution it proposes to solve this problem. The grades obtained by the application are in the range 7-10.

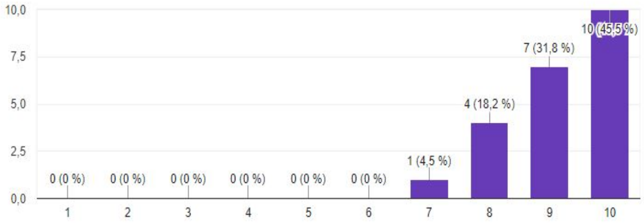


Figure 13: General opinion of the users interviewed

CONCLUSION

This paper describes an application capable of detecting the symptoms of a stroke. The app includes an F.A.S.T. digital, on the phone, so that it can be used by anyone. The F.A.S.T. it consists of 4 components: detection of facial asymmetry, detection of muscle weakness in the face, detection of speech problems and detection of muscle weakness in the hands.

The purpose of this application is to facilitate the diagnosis of stroke as soon as possible, in order to avoid possible problems involved in this disease. Currently, the duration of a stroke test using our application is about 2 minutes, but we intend to reduce this time even further in the future.

According to the reports and statistics made and presented in the chapter dedicated to testimonials, the integration of a narrator must be taken into account in the future, to guide users through audio instructions, through the application.

ACKNOWLEDGMENTS

This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/ H2020-SC1-2019-Two-Stage-RTD.

REFERENCES

1. Asteriadis, S., Nikolaidis, N., Nikolaidis, N., Pitas, I. A Review of Facial Feature Detection Algorithms. In book *Advances in Face Image Analysis: Techniques and Technologies*. 42-61 (2011), http://doi:10.4018/978 -1-61520-991-0.ch003
2. Barra, P., Bisogni, C., Nappi, M., Ricciardi, S. Fast QuadTree-Based Pose Estimation for Security Applications Using Face Biometrics. *Network and System Security. NSS 2018. Lecture Notes in Computer Science*, Springer, Cham. 11058 (2018) https://doi.org/10.1007/978-3-030-02744-5_12
3. Brown, M. M. Brain Attack: a new approach. *Clinical medicine (London, England)* 2(1): 60-5, (2002)
4. Clarke, D. J., Forster, A. Improving post-stroke recovery: the role of the multidisciplinary health care team. *Journal of Multidiscip Healthc*. 8: 433-442, (2015)
5. García, L., Tomás, J., Parra, L., Lloret, J. An m-health application for cerebral stroke detection and monitoring using cloud services. *International Journal of Information Management*, 45: 319-327, (2019)
6. Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8): 707-710, (1966)
7. Marin, C. Accidentul vascular hemoragic. *Doc.ro* (2019) https://doc.ro/accident-vascular-cerebral/accidentul-vascular-hemoragic
8. Neagu, A. 4 Romanians die every hour due to a stroke. Doctors are asking for funding and new centers to reduce the number of deaths. *Hotnews.ro* (2019)
9. Omer, Y., Sapir, R., Hatuka, Y., Yovel, G. What Is a Face? Critical Features for Face Detection. *SAGE Journals*, 48(5):437-446 (2019) https://doi.org/10.1177/0301006619838734
10. Rosebrock, A. Facial landmarks with dlib, OpenCV, and Python. *Pyimagesearch* (2017) https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/
11. Shugalo, I. How Artificial Intelligence Can Predict and Detect Stroke. *ITonline, Feature Artificial Intelligence*. (2019) https://www.itnonline.com/article/how-artificial-intelligence-can-predict-and-detect-stroke
12. Siriţeanu, A., Iftene, A. MeetYou - Social Networking on Android. *In Proceedings, of 11th International Conference RoEduNet IEEE: Networking in Education and Research*., Editor Rusu, O., 17-19 January 2013, Sinaia, Romania, 228-233, (2013)
13. Sorenson, T., Giordan, E., Lanzino, G. Neurosurgery for Ischemic and Hemorrhagic Stroke. *Fundamentals of Neurosurgery* (2019) https://doi.org/10.1007/978-3-030-17649-5_9
14. Ţucă, L. P., Iftene, A. Speech recognition in education. Voice Geometry Painter Application. *In the 9th Conference on Speech Technology and Human-Computer Dialogue*, *IEEE*, July 6-9, 2017, Bucharest, Romania, 1-8, (2017)
15. Weber, C. E. Stroke: Brain Attack, Time to React. *AACN Clinical Issues Advanced Practice in Acute and Critical Care* 6(4): 562-575, (1995)

Intrinsic motivation and motives for Facebook use – a formative measurement approach

Costin Pribeanu

Academy of Romanian Scientists

Str. Ilfov No.3, Bucharest, Romania

costin.pribeanu@aosr.ro

DOI: 10.37789/rochi.2020.1.1.8

ABSTRACT

For over a decade, Facebook is part of the everyday life of university students. Its growing popularity stimulated the research aiming to understand the motivations that are driving people to join and use social networking websites. To answer this research question various approaches have been taken: technology acceptance model (TAM), uses and gratification theory, the theory of consumption values, and multidimensional models. The objective of this paper is to analyze the relationship between the intrinsic motivation of using Facebook and the motives for its use. Intrinsic motivation for using Facebook has been conceptualized as perceived enjoyment. The motives for Facebook use have been conceptualized as a formatively-measured construct that impacts two drivers of the intention of use: the perceived enjoyment and the perceived ease of use. The results show that four motives for using Facebook are predicting Facebook's perceived enjoyment: keeping in touch with known people, entertainment, finding useful information and resources, and socialization.

Keywords

Motives for Facebook use, hedonic technologies, MIMIC models, perceived enjoyment, perceived ease of use, formative measurement.

ACM Classification

D.2.2: Design tools and techniques. H5.2 User interfaces.

INTRODUCTION

The growing popularity of Facebook among university students stimulated the research in the usage of social networking websites and raised a plethora of research questions that refer to: motives for Facebook use, usage characteristics (frequency and time spent daily, number of Facebook friends), Facebook acceptance and continued use, educational usefulness of Facebook, social learning, social influence, and problematic Facebook use.

Understanding the motives for Facebook use has been an important issue in social media research [1, 27]. A review of the extant literature shows various approaches to this research topic, including qualitative and quantitative studies. Several studies used the technology acceptance [32], theory of consumption values [1, 32, 33], social action theory [10] or uses and gratifications theory [10, 21] to conceptualize the variables of interest then tested the models and analyzed the results by using linear regression, analysis of variance, or structural equation modeling [1, 9, 21, 27]. Several approaches conceptualized the motives for Facebook use as multidimensional models [1, 10, 22].

Although the motives for Facebook use have been extensively researched, few studies are addressing the nature of motivation driving the intention to use.

The objective of this work is to analyze the relationship between the intrinsic motivation of using Facebook and the motives for Facebook use. In the framework of technology acceptance theory, the intrinsic motivation [12] has been conceptualized as perceived enjoyment, defined by Davis et al. [11] as "the extent to which the activity of using a specific system is perceived to be enjoyable in its own rights, aside from any performance consequences resulting from system use".

The motives for Facebook use could be grouped and conceptualized as reflectively measured dimensions of a multidimensional construct [1, 22]. However, the motives are quite diverse therefore a model addressing several dimensions requires a large number of constructs. A formative measurement approach has the advantage of using a set of indicators corresponding to the main categories of motives.

In this study, the motives for Facebook use have been conceptualized as a formatively measured construct that has an impact on two variables that are driving the intention to use: the perceived ease of use and the perceived enjoyment. In this respect, the study takes the perspective of consumption values theory [30]. This theory borrowed from marketing research has been used by Turel et al. [32] for the acceptance of hedonic artifacts, by Wang et al. [33] for the acceptance of mobile applications, and by Aldawani [1] for the conceptualization of motives for using Facebook.

The formative model has been tested on a sample of 182 Romanian university students.

The rest of this paper is organized as follows. In section 2, related work is discussed with a focus on the motives for using Facebook. The method and results of the empirical study are presented in sections 3 and 4. The paper ends with a conclusion in section 5.

RELATED WORK

Motives for Facebook use

The motives for using social networking websites, in general, and Facebook, in particular, have been studied from different theoretical perspectives by using both qualitative and quantitative approaches. Ellison et al [15] found that keeping in touch with friends and maintaining social relations are important motives for using Facebook by college students.

In the study of Park et al. [27], four primary needs for joining Facebook groups have been identified that vary by gender, hometown, and year in school: self-status seeking, socialization, entertainment and, and information.

Alhabash & Ma [3] analyzed the motivations of use among college students on four platforms: Facebook, Twitter, Instagram, and Snapchat. They found that the main motives for Facebook use are convenience, entertainment, passing time, medium appeal, and information sharing.

The study of Toker & Baturay [31] analyzed the factors that influence the use of Facebook for educational purposes. They found that students perceived Facebook as a good tool for communication and collaboration. The most influential factors were the GPA and personal use of Facebook for studying.

A previous study exploring the motives for Facebook use by Romanian university students [25] found that the main reasons are: communication with friends keeping in touch with former high school colleagues, and finding out what is new. Another study [5] analyzed ten motives for Facebook use and found that the most important was to communicate with friends, to find out what happens in university, and to keep in touch with former colleagues.

Iordache & Pribeanu [22] explored the motives for Facebook use from an educational perspective. They took a multidimensional approach by validating three dimensions: extending social relationships, information and collaboration, and maintaining social relationships. The study found that students are using Facebook mainly for maintaining social relationships.

The study of Cheung et al. [10] took the perspective of social action theory to explain the use of social networks by students. They conceptualized a model by considering the following factors: social influence, social presence, and five key values from uses and gratifications theory (purposive value, social enhancement, self-discovery, interpersonal interconnectivity, and entertainment value).

Ifinedo [21] analyzed the motives for Facebook use from the perspective of uses and gratifications theory. He found that the entertainment value and maintaining interpersonal relationships have the most important influence on the behavioral intention to use.

Hedonic technologies

The theory of consumption values states that consumer choice is a function of multiple consumption values that are independent and make differential contributions in different choice situations [30]. They identified five key values: functional, conditional, social, emotional, and epistemic. According to the authors, the theory may be used to predict, describe, and explain the consumer's behavior.

Hirschman & Holbrook [19] defined the hedonic consumption in terms of "consumers' multisensory images, fantasies, and emotional arousal in using a product". In their review of pleasure principles, Alba & Williams [2] distinguished pleasure in the product (aesthetics, design, having vs. doing) and pleasure from person-product interaction (from expectations and engagement).

Hassenzahl [17] analyzed the hedonic quality as different from the ergonomic quality in the context of human-computer interaction. He distinguished between pragmatic (utility, usability) and hedonic (identification, stimulation, and evocation) attributes of a product.

Heijden [18] analyzed the differences in technology acceptance models for utilitarian and hedonic systems. As he pointed out, "the value of a hedonic system is a function of the degree to which the user experiences fun when using the system".

Turel et al. [32] analyzed the acceptance of hedonic technologies from the perspective of consumption values theory. They conceptualized the overall hedonic value as a third-order factor model having on the second level four key values: appeal value, social value, playfulness value, and value for money.

Based on a literature review, Aldawani [1] identified four categories of motives for using Facebook (termed as major facets of Facebook gravitation): social, functional, emotional, and, epistemic. Then, based on principal component analysis, he identified eight factors related to the motives for using Facebook: connecting (creating, developing, and maintaining relationships), sharing content, relaxing, branding, organizing (meetings and events), monitoring (friends, celebrities, colleagues), expressing oneself, and learning. The multidimensional approach resulted in a 34-item evaluation instrument. The study of Aldawani is rooted in the consumption values theory applied to the analysis of hedonic technologies.

Wang et al. [33] used the theory of consumption values to analyze the use of mobile applications. In their model, the intention to use is driven by four key values: functional, social, emotional, and epistemic. The results showed that emotional and epistemic values had the strongest influence.

Yang and Lin [34] investigated the motives that influence the stickiness to Facebook from the perspective of a value-based theory. Their model includes a moderating variable (trust in Facebook) and three values: social value, epistemic value, and hedonic value. The results showed that the hedonic value is positively influencing the stickiness for Facebook. A group analysis showed that in a high-trust group social and hedonic values have a significant impact while in a low-trust group epistemic and hedonic values are significant predictors of the stickiness to Facebook.

Formative measurement models

In information systems research, a distinction is made between two types of model: measurement model and structural model. The former describes the causal relationships between a latent variable (construct) and its measures (indicators, items, observed variables) while the latter describes the causal relationships between latent variables. According to Anderson and Gerbing [4], before estimating and assigning semantics to the structural model the measurement model has to be correctly specified.

An important issue is the correct specification of the measurement model [4, 6, 23].

Following the direction of causal relationships, there are two types of measurement model: reflective and formative (see Figure 1), having distinct characteristics. In the reflective measurement model, the causal direction is from the latent variable to indicators so a change in the latent variable is reflected in simultaneous changes in all indicators. Indicators should be positively correlated and the measurement model should have convergent validity [6].

In formative measurement, the causal direction is from indicators to construct. Indicators are not interchangeable since each is capturing a distinct cause. There are no assumptions on unidimensionality or correlations between indicators. Indicators don't have an error term and items are intercorrelated [7, 8, 13, 14].

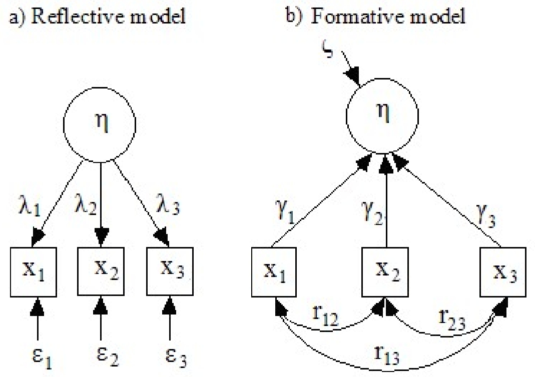


Figure 1. Reflective and formative measurement models

A formative measurement model taken in isolation is under-identified and cannot be estimated. Most authors recommend achieving identification based on the specification of effects (outcomes) on at least two unrelated variables that are reflectively measured. The effect variables could be two reflective indicators (MIMIC model), two reflective constructs, or a reflective construct and a reflective indicator.

METHOD

Research model and measures

The motives for using Facebook (FBU) are influencing two variables: the perceived ease of use (PEU) and the perceived usefulness (PU). The research model is presented in Figure 1.

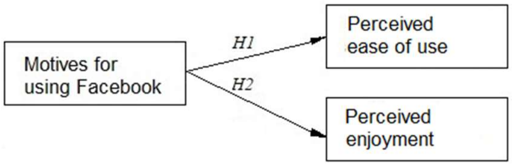


Figure 1. The research model

The following two hypotheses are tested in this study:

[H1] The motives for using Facebook have a positive influence on the perceived ease of use (FBU → PEU).

[H2] The motives for using Facebook have a positive influence on perceived enjoyment (FBU → PE).

In this research, six categories of motives have been considered: meeting new people, keeping in contact with known people, finding information and resources, socialization, collaboration, and entertainment. The construct "Motives for using Facebook" (FBU) has been conceptualized as a formatively measured construct, having six indicators.

The choice of indicators is based on the results of previous studies [5, 22, 25].

The research model is operationalized as a MIMIC model [24]. MIMIC model is the simplest formative model having multiple indicators (reflectively measured) and multiple causes (formatively measured) of a single latent variable. The variables used in this study are presented in Table 1.

Table 1. Variables

FBU1	I use Facebook to get in touch with new people
FBU2	I use Facebook to keep in touch with people I know
FBU3	I use Facebook to find information and resources
FBU4	I use Facebook for socialization purposes
FBU5	I use Facebook for collaboration purposes
FBU6	I use Facebook for entertainment purposes
PEU	Facebook is easy to use
PE	I like to use Facebook

Additionally, two regression models have been tested, in order to analyze the effect of formative indicators on each outcome variable.

Validation criteria

The following criteria have been used to assess the validity of the model: coverage of the domain of content, correct sign and significance of γ-coefficients, significant influence on the outcome variables (λ-coefficients), and an acceptable fit of the model with the data [8, 14]).

Based on the recommendations from the literature [16, 20, 29], the following goodness-of-fit measures were used: chi-square (χ^2), normed chi-square (χ^2/df), comparative fit index (CFI), goodness-of-fit index (GFI), standardized root mean square residual (SRMR), and root mean square error of approximation (RMSEA).

The formative measurement model was analyzed with Lisrel 9.3 for Windows [26], using a covariance matrix as input and maximum likelihood estimation method

EMPIRICAL STUDY

Sample and data analysis

The questionnaire has been administrated in May 2019. A total of 194 students from the University of Building Engineering in Bucharest participated in the study. The students have been asked to answer general questions such as demographics (age, gender), enrollment (university, faculty, year of study), FB usage (size of their FB network, frequency of use, minutes per day), then to evaluate items on a 7-points Likert scale.

A total of 12 questionnaires have been eliminated for incomplete data so the working sample has 182 observations (127 male and 55 female). The age of participants ranges between 18 and 34 years (M=20.36, SD=2.00).

Model estimation results

The model estimation results are presented in Figure 3

The goodness of fit indices (GOF) indicate a very good level of fit of the proposed model with the data: χ^2=11.76, DF=5, p=0.038, χ^2/DF=2.352, CFI=0.976, GFI=0.985, SRMR= 0.034, RMSEA=0.086.

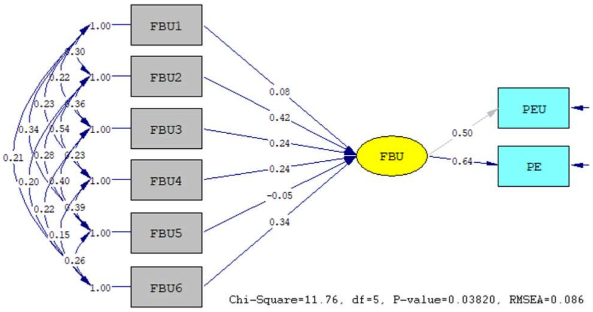


Figure 3. Model estimation results

The descriptive statistics, the influence of the focal construct on the outcomes (λ), and γ-coefficients are presented in Table 2. With one exception (FBU1), the observed scores are over the neutral value of 4.00. The indicators FBU2, FBU4, and FBU6 have the largest mean values. The influence of FBU on PEU (β=0.50) and PE (β=0.64) is significant at p<0.001 level, which supports the two hypotheses H1 and H2. The model explains a 76.3% variance in the focal construct. The error term (error variance of FBU) is only 0.237 which shows good coverage of the domain of content.

Table 2. Descriptive statistics, loadings(λ), and γ-coefficients

Item	M	SD	λ	γ	Sig.
FBU1	3.54	1.60		0.08	0.850
FBU2	5.69	1.55		0.42	0.001
FBU3	4.22	1.75		0.24	0.000
FBU4	5.18	1.66		0.24	0.009
FBU5	4.17	1.76		-0.05	0.034
FBU6	5.13	1.73		0.34	0.005
PEU	6.10	1.44	0.50		0.000
PE	4.31	1.66	0.64		0.000

One indicator has a negative γ-coefficient (FBU5) and another indicator has nonsignificant γ-coefficient (FBU1). The formative indicators having the largest γ-coefficients are FBU2 (γ = 0.42, p=0.000) and FBU6 (γ = 0.34, p=0.000). The other two indicators, FBU3 and FBU4 are significant at p<0.05 level. The correlations between indicators are not too high, bellow the recommended threshold value [14].

The incorrect sign and the lack of significance of FBU1 and FBU5 suggest that these are not valid measures of the focal construct [8, 14] and therefore might be eliminated. The results of testing the revised model are presented in Figure 4.

All four indicators are significant. The largest contribution to the focal construct is given by FBU2 (γ = 0.44, p=0.000) which shows that keeping in touch with known people has been perceived as the most important motif. Next indicators in the order of importance are FBU6 (γ = 0.34, p=0.000), FBU3 (γ = 0.24, p=0.011), and FBU4 (γ = 0.23, p=0.021).

Overall, the model explains 75.9% variance in the motives for using Facebook, 40.7% in the perceived enjoyment, and 24.9% in the perceived ease of use.

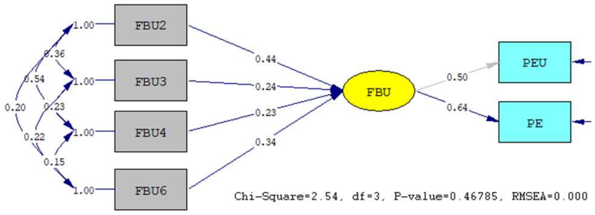


Figure 4. Revised model estimation results

The GOF indices of the revised model are also indicating an excellent fit of the model with the data: χ^2=2.54, DF=3, p=0.468, χ^2/DF=0.836, CFI=1.000, GFI=0.995, SRMR= 0.018, RMSEA=0.000.

Regression analysis

In order to extend the analysis, two regression models have been tested having the perceived enjoyment (PE) and perceived ease of use (PEU) as dependent variables and the six FBU indicators as predictors (independent variables). The results are presented in Table 3 and Table 4.

Table 3. Regression analysis results for PE

PU	β	Error	t-Stat	p-value
Intercept	-0.04	0.49	-0.08	0.937
FBU1	0.11	0.07	1.52	0.131
FBU2	0.23	0.08	2.70	0.008
FBU3	0.14	0.07	2.00	0.047
FBU4	0.14	0.08	1.79	0.075
FBU5	0.06	0.07	0.81	0.420
FBU6	0.22	0.06	3.52	0.001

For PE, the multiple correlation (R=57.10) is significantly different from 0, F (6, 181) = 14.11, p=0.000, with adjusted R^2=32.61%. The largest influence on PE have FBU2 (β=0.22, p=0.008) and FBU6 (β=0.22, p=0.001). The regression explains a 30.3% variance in the perceived enjoyment.

Table 4. Regression analysis results for PEU

PEU	β	Error	t-Stat	p-value
Intercept	3.38	0.455	7.44	0.000
FBU1	-0.03	0.07	-0.48	0.634

FBU2	0.29	0.08	3.61	0.000
FBU3	0.12	0.06	1.88	0.062
FBU4	0.13	0.07	1.76	0.081
FBU5	-0.14	0.07	-2.13	0.035
FBU6	0.12	0.06	2.10	0.037

For PEU (Table 4), the multiple correlation is R=46.98, F (6, 181)= 8.26, p=0000, with adjusted R^2=22.07%. The largest influence has FBU2 (β=0.28, p=0.000), then FBU4 (β=0.13, p=0.081), and FBU3 (β=0.12, p=0.081). Two coefficients (FBU3 and FBU4) are only marginally significant.

The regression explains 19.40% variance in the perceived ease of use.

Discussion

The results of this study show that the main reasons why university students Facebook are: keeping in touch with people they know (maintaining social relationships), finding useful information and resources, socialization, and entertainment.

The results are confirming previous findings [22, 25] and are congruent with the results of many studies as regards the social and hedonic value of Facebook [9, 15, 21, 27].

The model explains a 40.7% variance in the perceived enjoyment. This is not surprising, given the hedonic nature of Facebook [27]. As the results of this study show, three out of four significant motivators for using Facebook have social value (keeping in touch with known people and socialization) and hedonic value (entertainment). As many authors pointed out [10, 15, 34], social activities on Facebook are exciting thus enhancing its perceived hedonic value. Also, for specific social groups (university students, first-year students) socialization itself is enhancing the hedonic value [19, 35].

The model estimation results as well as the regression results confirm that two indicators are not suitable. The results are pretty similar as regards the influence of motivators on the two outcome indicators. On the other hand, some differences exist that are explained by the nature of the model.

The regression analyses highlight the particulat relevance of indicators for the outcomes variables. In this respect, FBU2 (keeping in touch with known people) is more relevant for the perceived ease of use than for the perceived usefulness. This suggests that the tasks of keeping in touch with known people are perceived as requiring more effort. On the other hand, the influence of two indicators (FBU3 and FBU4) on the perceived ease of use is only marginally significant.

This work contributes to a deeper understanding of the motives for Facebook use and how these motives are mirrored in the perceived enjoyment and perceived ease of use. There is no quantitative study up to now addressing these research questions from a formative measurement perspective. The main advantage of this approach is an instrument having a small set of indicators. Second, it shows the advantage of using a mix of methods. The regression analyses enable a better explanation of the formative model estimation results and validation of the formative indicators.

There are several limitations of this exploratory study. First of all, the formative measurement has its own limitations related to model identification and validation. The number of formative indicators is small and two indicators have been eliminated. Future work should enlarge the set of motives for Facebook use.

As many authors pointed out [21, 27, 33], the motives for Facebook use depend on a diversity of factors. Therefore, the conclusions of this study using a sample of college students should not be generalized to other populations.

CONCLUSION AND FUTURE WORK

This study contributes to a better understanding of the intrinsic motivation for using Facebook by taking a formative measurement approach. The results show that perceived enjoyment is predicted by four main motives for Facebook use: keeping in touch with known people, finding useful information & resources, socialization, and entertainment.

Although the results are validating a small set of four indicators, representing four categories of motives, the model is explaining 76% of the variance in the latent variable, which shows good coverage of the formative indicators and suggests a promising starting point for future studies.

REFERENCES

1. Aladwani, A. M. (2014). Gravitating towards Facebook (GoToFB): What it is? and How can it be measured?. *Computers in Human Behavior*, 33, 270-278. http://dx.doi.org/10.1016/j.chb.2014.01.005
2. Alba, J. W., & Williams, E. F. (2013). Pleasure principles: A review of research on hedonic consumption. *Journal of consumer psychology*, 23(1), 2-18. https://doi.org/10.1016/j.jcps.2012.07.003
3. Alhabash, S., & Ma, M. (2017). A tale of four platforms: Motivations and uses of Facebook, Twitter, Instagram, and Snapchat among college students?. *Social Media+ Society*, 3(1), 2056305117691544. https://doi.org/10.1177/2056305117691544
4. Anderson, J.C. & Gerbing, D.W. (1988) Structural Equation Modeling in Practice: A Review and Recommended Two-Step Approach. *Psychological Bulletin*, 103(3), 411-423.
5. Balog, A., Pribeanu, C. Ivan, I. (2015) Motives and characteristics of Facebook use by students from a Romanian university. In: Dardala, M., Rebedea, T.E. (Eds.) *Proceedings of RoCHI 2015*, Bucharest, 24-25 September, 137-140.
6. Bagozzi, R. P. (2011). Measurement and meaning in information systems and organizational research: methodological and philosophical foundations. *MIS Quarterly*, 35(2), 261–292.
7. Bollen, K. A., & Diamantopoulos, A. (2017). In defense of causal-formative indicators: A minority report. *Psychological Methods*, 22(3), 581. DOI: 10.1037/met0000056

8. Boolen, K. (2011) Evaluating effect, composite and causal indicators in structural equation models. MIS Quarterly, 35(2), 359-372.
9. Chang, C. C., Hung, S. W., Cheng, M. J., & Wu, C. Y. (2014). Exploring the intention to continue using social networking sites: The case of Facebook. Technological Forecasting and Social Change, 95, 48-56. https://doi.org/10.1016/j.techfore.2014.03.012
10. Cheung, K., Chiu, P.-Y., Lee, M. (2014) Online social networks: Why do students use Facebook. Computers in Human Behavior, 27(4), 1337-1343. https://doi.org/10.1016/j.chb.2010.07.028
11. Davis, F.D., Bagozzi, R.P., Warshaw, P.R. (1992). Extrinsic and intrinsic motivation to use computers in the workplace. Journal of Applied Social Psychology, 22 (14), 1111-1132.
12. Deci, E. L., Vallerand, R. J., Pelletier, L. G., & Ryan, R. M. (1991). Motivation and education: The self-determination perspective. Educational psychologist, 26(3-4), 325-346.
13. Diamantopoulos, A., &Winklhofer, H. (2001). Index construction with formative indicators: an alternative to scale development. Journal of Marketing Research, 38(2), 269–277.
14. Diamantopoulos A. (2011) Incorporating formative measures into covariance-based structural equation models. MIS Quarterly, 35 (2). 335-358.
15. Ellison, N.B., Steinfield, C., Lampe, C. (2007). The benefits of Facebook "Friends:" Social capital and college students' use of online social network sites, Journal of Computer-Mediated Communication, 12, 1143-1168
16. Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E., Tatham, R.L. (2006). *Multivariate Data Analysis*. 6th ed., Prentice-Hall.
17. Hassenzahl, M. The Thing and I: Understanding the Relationship Between User and Product. In Blythe, M., Overbeeke, K., Monk, A., and Wright, P. (Eds.). *Funology: From Usability to Enjoyment.* Kluwer Academic Publishers, 2005, 31–42
18. Heijden, H. van der (2004). User acceptance of hedonic information systems. *MIS Quarterly*, 28 (4), 695-704.
19. Hirschman, E. and Holbrook, M.B. (1982), "Hedonic consumption: emerging concepts, methods and propositions", *Journal of Marketing*, Vol. 46 No. 3, pp. 92-101
20. Hu, L. T., & Bentler, P. M. (1998). Fit indices in covariance structure modeling: Sensitivity to under parameterized model misspecification. *Psychological methods*, 3(4), 424.
21. Ifinedo, P. (2016). Applying uses and gratifications theory and social influence processes to understand students' pervasive adoption of social networking sites: Perspectives from the Americas. International *Journal of Information Management*, 36(2), 192-206. http://dx.doi.org/10.1016/j.ijinfomgt.2015.11.007
22. Iordache, D. D., & Pribeanu, C. (2016). Exploring the motives of using Facebook – a multidimensional approach. *Revista Romana de Interactiune Om-Calculator*, 9(1), 19-34.
23. Jarvis, C.B., Mackezie, S., Podsakoff, M. (2003) A critical review of construct indicators and measurement models misspecification in marketing and consumer research. Journal of Consumer Research 30, 199-218.
24. Jöreskog, K. G., & Goldberger, A. S. (1975). Estimation of a model with multiple indicators and multiple causes of a single latent variable. *Journal of the American Statistical Association*, 70(351a), 631-639.
25. Manea, V.I., Gorghiu, G., Iordache, D.D. (2015) The educational potential of Facebook use by students in two Romanian universities*. Revista Romana de Interactiune Om-Calculator* 8(3), 195-208.
26. Mels, G. (2006). *LISREL for Windows: Getting Started Guide*. Lincolnwood: Scientific Software International, Inc.
27. Park, N., Kee, K. F., Valenzuela, S. (2009). Being immersed in social networking environment: Facebook groups, uses and gratifications, and social outcomes. *CyberPsychology & Behavior*, 12(6), 729-733.
28. Rosen, P., Sherman, P. (2006) Hedonic information systems: Acceptance of social networking websites. *Proceedings of AMCIS 2006*, Acapulco, 4-6 August, 1218-1223.
29. Schermelleh-Engel, K., Moosbrugger, H., & Müller, H. (2003). Evaluating the fit of structural equation models: Tests of significance and descriptive goodness-of-fit measures. *Methods of psychological research online*, 8(2), 23-74
30. Sheth, J. N., Newman, B. I., & Gross, B. L. (1991). Why we buy what we buy: A theory of consumption values. *Journal of Business Research*, 22(2), 159-170.
31. Toker, S., & Baturay, M. H. (2019). What foresees college students' tendency to use Facebook for diverse educational purposes?. *International Journal of Educational Technology in Higher Education*, 16(1), 9. https://doi.org/10.1186/s41239-019-0139-0
32. Turel, O., Serenko, A., & Bontis, N. (2010). User acceptance of hedonic digital artifacts: A theory of consumption values perspective. *Information & Management*, 47(1), 53-59.
33. Wang, H. Y., Liao, C., & Yang, L. H. (2013). What affects mobile application use? The roles of consumption values. International Journal of Marketing Studies, 5(2), 11-22, DOI:10.5539/ijms.v5n2p11
34. Yang, H. L., & Lin, C. L. (2014). Why do people stick to Facebook web site? A value theory-based view. *Information Technology & People*, 27(1), 21-37. http://dx.doi.org/10.1108/ITP-11-2012-0130.
35. Yang, C. C., & Brown, B. B. (2013). Motives for using Facebook, patterns of Facebook activities, and late adolescents' social adjustment to college. *Journal of youth and adolescence*, *42*(3), 403-416. DOI 10.1007/s10964-012-9836-x.

.

Targeted Romanian Online News in a Mobile Application Using AI

Marius-Cristian Buzea
University Politehnica of Bucharest, Department of Computer Science and Engineering, Bucharest, Romania
bumarius@gmail.com

Ştefan Trăuşan-Matu
University Politehnica of Bucharest, Department of Computer Science and Engineering, Bucharest, Romania
stefan.trausan@cs.pub.ro

Traian Rebedea
University Politehnica of Bucharest, Department of Computer Science and Engineering, Bucharest, Romania
traian.rebedea@cs.pub.ro

DOI: 10.37789/rochi.2020.1.1.9

ABSTRACT

Nowadays, being informed is a very important aspect of our life. Few online news broadcasters, such as bbc.com or cnn.com, send real-time mobile notifications with several interactivity facilities with the application to be informed with the latest news. In this paper, we present an application based on Machine Learning (ML), a basic technique used in Artificial Intelligence (AI), available for mobile devices, which detects irony, the articles' polarity and the fake news. The mobile application is based on a machine learning system using manually annotated Romanian corpora, and a crawler network designed to automatically extract data from more than 730 public websites. The aim of the application is to collect information regarding user behavior, to change and adapt the content to user's preferences and requirements in order to process and better understand the user's necessities. Furthermore, the user is able to search through the news, filter the news, create favorites lists and visualize graphical representations for different time periods. Therefore, the main goal of the application is to develop and improve the way that people receive and access reliable information.

Author Keywords

Adaptive Mobile News Platform; Artificial Intelligence; Machine Learning; Romanian Corpora; Graphical Representations;

ACM Classification Keywords

H.5.2: User Interfaces.

INTRODUCTION

Nowadays, the number of mobile devices surpasses 3 billion and is estimated[1] to grow with more than 1 billion in the next few years. Recent statistics show that the global mobile data traffic is increasing every year with 30%[2], this aspect being illustrated by Figure 1, which presents the traffic data from a Google analytics account belonging to a website with more than 2.5 million pages visited per year. The rising number of mobile applications is generating a significant amount of useful data, which can help a company or a public person with online business strategies. At the same time, this phenomenon can have a negative impact for some business if the data collected cannot be selected and filtered by a specific target or topic. Moreover, analyzing and achieving graphical representations of statistical data for different time periods give to users the opportunity to take measures in case of the occurrence of unpleasant situations.

How and when people use their mobile devices is another aspect involved before starting to design and develop the application, because the mobile users often read news on their gadgets in the car or subway. In general, when creating new attractive applications, most programmers adopt the same principles that are normally applied to the web interfaces, but it is known that the issues are more pronounced due to the limited mobile space, a solution being the simplicity. The display is narrow or the network connectivity is slower, but what we lose in presentation layer, we gain in having a carefully designed user interface or better interactions with the technology, having gadgets with touch screen being more natural [1].

There are several Romanian news mobile platforms available in Google store and App store, such as "Pressa - Știri din România" (https://www.pressa.app/), DCNews.ro or "Observator" (https://observatornews.ro/) but these applications do not have advanced integrated facilities, such as fake news detection, ironic news detection or negative news detection.

Most of the existing applications extract data from different sources using public RSS feeds, browser extensions or poor scrapers. Moreover, what really matters is the number of followers or the number of subscribers on social media platforms. Because of these rules, the online publishers require their employer's exclusivity and the above sources, most of the time, are not immediately updated.

[1] https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, last accessed on 5th July 2020

[2] https://mylittlebigweb.com/en/should-you-switch-to-amp-format-for-better-seo-performance/, last accessed on 5th July 2020

Our developed mobile platform offers a solution to these issues, by using Machine Learning (ML), a basic technique used in Artificial Intelligence (AI) for the extraction process to achieve high performances. Furthermore, the application is adapting the content to the user's preferences and necessities by registering several keywords for every topic or public target.

Another important aspect of our platform is represented by the content, such as images, texts and videos which are stored in our application. The known applications store only the title of the news with a link through the original source and, in most of the cases, when the online publishers delete the information from their website, the user cannot access that source anymore.

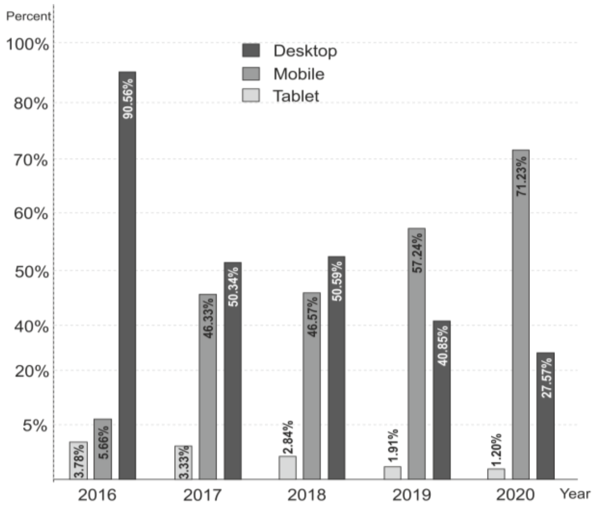


Figure 1: Comparative data traffic

The datasets used in machine learning processes to identify the news polarity, the ironic, fake or negative news in Romanian online broadcasters are another important aspect of this paper. The datasets are based on a Romanian language dictionary, a corpus of non ironic news and ironic news items which were manually annotated by employees in a public institution from Romania.

One of the objectives of this paper is to describe the module for automatic data selection of the mobile application, based on web page properties. This module is used in further extraction mechanisms, representing the main element of the machine learning system. To achieve content for our platform, we used keywords and html selectors, such as "div", "meta tags" or "img".

This paper is also presenting mobile platform techniques, which, alongside machine learning algorithms, might be useful in using these architectures in future mobile application developments.

The paper is organized as follows: next section presents the status and trends related to the developed application; the following section describes the method and main components of the platform; next, the detailed architecture is presented, followed by the user interface section; finally, last section presents the conclusions of our work.

STATUS AND TRENDS

With so many smartphone users across the world, who could download in 2019 between 2.6 million Android and 2.2 million iOS applications, it is no surprise that the mobile applications industry is expanding [2]. The Apple App Store [3] and the Google Play Store [4] dominate this field, being the most important players in domain. Several applications are free and others have a price, the profit being split between the application's creator and the distribution platform. App usage and mobile devices are still growing [5] at a constant rate and it shows no signs of slowing down. We use our devices at home, in subway and even in our bed, spending most of the time on applications, such as Facebook, Twitter or Instagram. These social platforms are adaptable, innovative [6] and have contributed to our way of communication, creating a way for people to update and share content with their friends and even to learn how these can be used in the business online market campaigns.

Mobile devices offer now more facilities than desktops, due to various functionalities that detect the location or surrounding places, the portability and attractive mobile user interfaces [7]. Compared to desktop user interface, the mobile user interface design requirements [8, 9], are different for every application to ensure usability, consistency, and readability [10]. Due to the smaller screen, the mobile interface [11, 12] focuses on efficiency and discoverability, therefore the user should be able to understand a command through the icons, such as trash or heart pictures, defining the delete action or adding to favorite list.

There are several Romanian applications which follow the above requirements being carefully designed and developed, such as Google News, Biziday (https://www.biziday.ro/) and Digi24 (https://www.digi24.ro/), designed to receive updates in real time from different public sources, which were the most downloaded Romanian online news applications in 2019.

METHOD

For the past several years, the Google Play Awards have recognized the top applications for Android. The nominees were selected by various teams across Google for nine different categories. These applications should provide the best experience for users, interactive user mobile interfaces, new technologies, but these platforms are not delivering information such as news to the communities. We consider that there is still enough space for the development of these types of applications in online media. Most people would not even consider getting a physical morning newspaper anymore, so they need to use digital sources for daily news. The developed application presented in this paper tries to help users from the Romanian environment to get useful and reliable information depending on their interest topics, categories, targets or preferences.

Our application has an additional number of new features integrated than the above specified platforms, such as fake and ironic news category, a polarity category or a public targets category.

This application is based on the following components:

- The machine learning component.
- The core and backend web application components.
- The mobile user interface component.

The designed platform serves up news updates from a variety of sources giving to user the opportunity to receive or to be notified with the most relevant news. Moreover, the user is able to create favorite lists or other lists with certain publications saved to be read later.

ARCHITECTURE

Figure 2 illustrates the three main components of our platform, each of them supporting the system in different phases: from the machine learning process to the user interfacing.

In the first component, the machine learning system's components are integrated using PHP, MySQL and Python scripts, constituting the environment that allows the next processes to use and develop the interface between the AI modules and human users, between generated data and user's preferences. This component is based on more than 25,000 items news and a Romanian language dictionary containing more than 42,000 words. To evaluate the news polarity is used a semi-supervised machine learning system based on a three word-level approach [13]. To detect the irony in the online news for the Romanian language is used a corpus based on 14,064 ironic news items from the online environment based on automatic irony detection approach [14].

Another important aspect of this application is the extraction process which is based on more than 730 websites whose properties were manually annotated. After having added new websites to the application, the system evaluates the html properties by searching similar patterns with the best fit from database, to create articles for the latest inserted domain. Otherwise, when the website has not valid properties and the pages cannot be scanned, the platform requires human intervention. The bots network extract and expose only public content from REST API plugin or robots.txt file for the purpose of creating new value from data. Furthermore, it is provided a User Agent string that reveals the application's intention and the requested data based on DOM elements, such as div, are performed at a reasonable rate to consume minimum hardware and bandwidth resources to be never confused with a DDOS attack.

The second component provides the mandatory processes, generating the necessary data by the application core component, designing the connection between UI (User Interface) and AI. In the web application component, several keywords are manually added, to define every target or company/institution. Two alternative ways can be used: an automated process, when the system generates all possible vocabulary combinations of words, or a manual process, when the development team registers only the keywords of interest, gaining more speed for the machine learning processes. When the crawler's network detects some of these words in the news' content, the achieved data are passed to the next step, where several filtered functions and a sentiment analysis approach [13] are applied to remove ads and to set the polarity level.

The third component includes other important functionalities, such as favorite lists or saved for later read lists, which are a part of the mobile user interface component based on users' requirements, making the experience more attractive. In this field it is very difficult to distinguish a standard set because the mobile industry is evolving fast, giving the applications' authors free imagination finding and developing new components.

This component is implemented using the Ionic Framework language, making the developed platform available for both mobile operating systems Android and iOS.

It is not mandatory to use the three modules sequentially. When we have made changes to web application module, the bots network is updating, the machine learning system is changing, but the application core and user interface modules remain unchangeable and accessible.

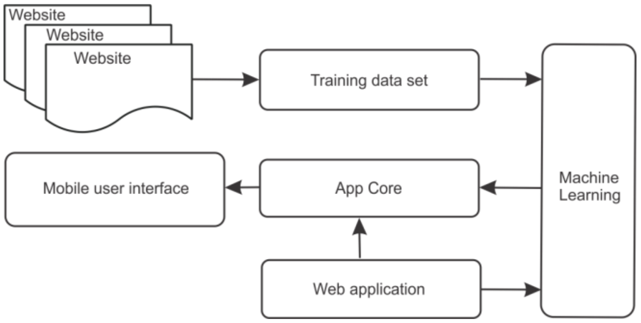


Figure 2: The architecture of the application

USER INTERFACE

We consider that the evolution of human beings in the last few years has been influenced by mobile technologies. Usage rates for those aged from 25 to 65, which are the main target group for this application, increased every year. The target population usually has varying characteristics and necessities, depending on cultural level or the capacity of the people to understand certain concepts. We have to develop elements for the mobile platform and place items or icons in ways that the user can take some desired actions or to interpret some information from graphical representations. Furthermore, we have to generate information that is easy to access and quickly to share. These are several aspects of the platform that transform the mobile user interface in the most important component of our system.

The developed application has the following features: viewing and selecting news for public targets or companies, selecting the negative, fake or ironic news, searching through news, filtering by company, sharing news, sorting news, adding to the favorite lists, editing users' information or viewing the graphical representation for a selected period. The user must be authenticated to access these features. He/she can require a new account and the credentials are analyzed by the development team.

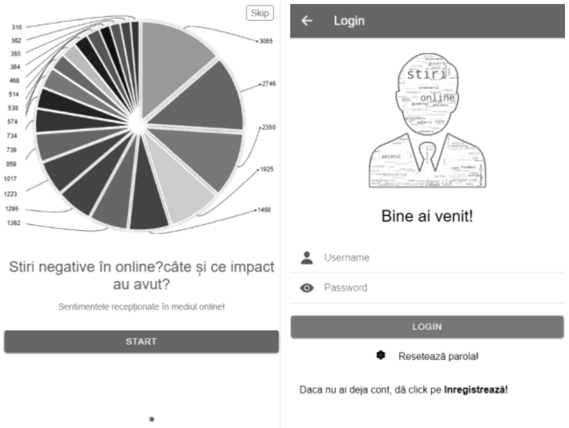


Figure 3: (a) Start page. (b) Login page.

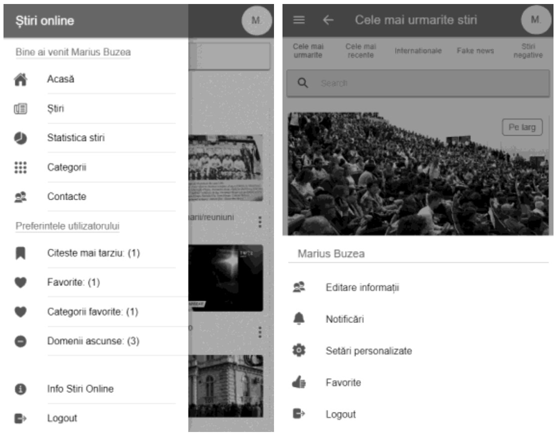
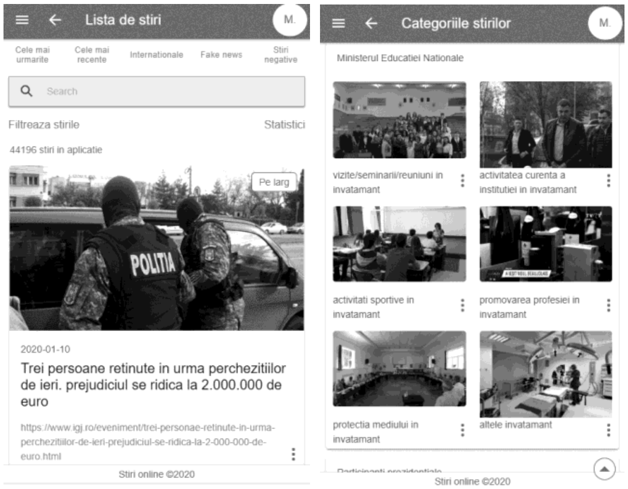


Figure 4: (a) Users' preferences. (b) Users' settings.

Figure 3 (a) presents the start page and Figure 3 (b) the login page, where the user has to choose between username and email address to get access in the developed platform. If the user forgets the password or the account was hacked, he/she can reset the password by pressing the button "Reseteaza parola" (eng., "Password reset"). After this action is completed, the user receives a link through email address to change the password.

After the registered user is logged, the recent stories are displayed with several options in the top of the page, giving to user the opportunity to select different categories. Also, the settings icon is positioned in the top left corner, as shown in Figure 4 (a) and (b). By pressing this button, the navigation menu is expanded. The user finds shortcuts to the main facilities of the platform, such as news of the day (Figure 5 a), news categories (Figure 5 b), statistics, graphical representation of the selected news, topics, useful contacts, information about this application and the most important aspect, the users' preferences category.

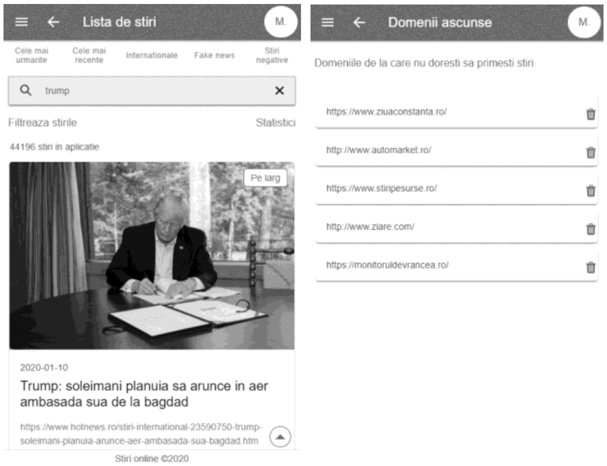


2020-01-10

Trei persoane retinute in urma perchezitiilor de ieri. prejudiciul se ridica la 2.000.000 de euro

https://www.igj.ro/eveniment/trei-personae-retinute-in-urma-perchezitiilor-de-ieri-prejudiciul-se-ridica-la-2-000-000-de-euro.html

Figure 5: (a) News list. (b) News categories.



2020-01-10

Trump: soleimani planuia sa arunce in aer ambasada sua de la bagdad

https://www.hotnews.ro/stiri-international-23590750-trump-soleimani-planuia-arunce-aer-ambasada-sua-bagdad.htm

Domeniile de la care nu doresti sa primesti stiri

https://www.ziuaconstanta.ro/

http://www.automarket.ro/

https://www.stiripesurse.ro/

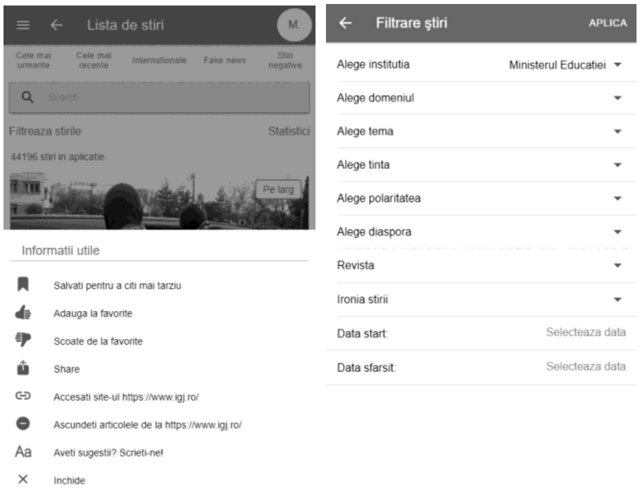
http://www.ziare.com/

https://monitoruldevrancea.ro/

Figure 6: (a) Search option. (b) Hidden domains.

The idea of designing and developing this application came from the necessity of a better connection between users and the online environment, in order to prevent the spread of fake news and misleading stories. The users' requirements category was implemented to ban the online broadcasters of false information or online deepfakes without proper disclosures. This category contains items, such as those saved for read later, favorite lists or hidden domain lists (Figure 6 b), giving the user the opportunity to refute certain domains and receive reliable information. Moreover, by creating the hidden domain lists, the application receives the necessary feedback, thus the programmers are able to improve or to be more carefully in the selection process for the specified broadcasters.

The search facility (Figure 6 a) of the application allows the user to find a specific story from the lists, providing further details such as image, description and more option (Figure 7 a) with link through the extended story or to the original source.



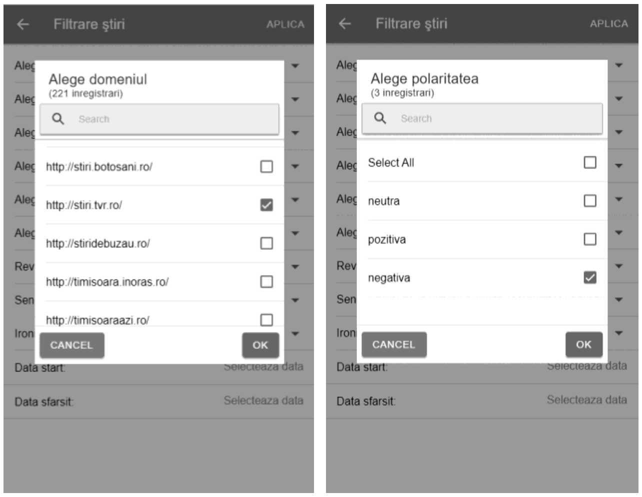
44196 stiri in aplicatie

Alege institutia: Ministerul Educatiei

Data start: Selecteaza data

Data sfarsit: Selecteaza data

Figure 7: (a) More options. (b) Filter options.

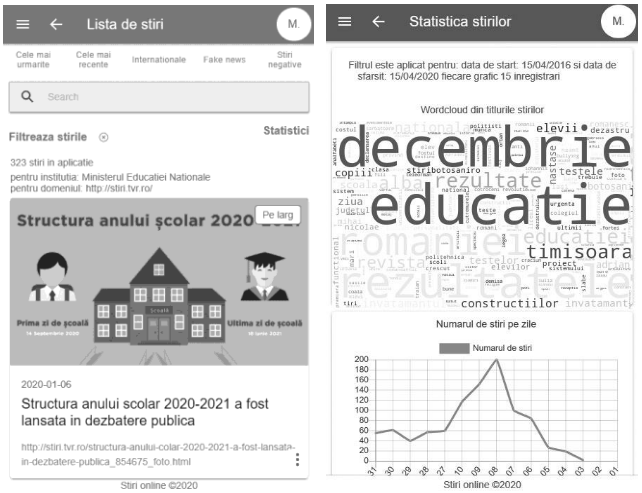


□ http://stiri.botosani.ro/
☑ http://stiri.tvr.ro/
□ http://stiridebuzau.ro/
□ http://timisoara.inoras.ro/
□ http://timisoaraazi.ro/

□ Select All
□ neutra
□ pozitiva
☑ negativa

Figure 8: (a) Domain options. (b) Polarities options.

One of the most important facilities of the designed platform is the filter module. The data from this module were manually designed by the management team, choosing relevant keywords for every institution or public target (ex. “ministru educatie” belonging to “Ministerul Educatiei si Cercetarii”). After having added the words in application, the system automatically detects all possible combinations of words, such as “ministrul educatiei”, “ministrilor educatiei”, “ministrului educatiei”, etc. Moreover, these are registered into the database to achieve a better targeting process. As illustrated in Figure 7 (b), the platform has several filtering options, such as topic, target or company/institution which have associated different domains (Figure 8 a). Furthermore, to gain speed and have a better search experience, we initially loaded only 15 domains, so that the user has to scroll vertically to load more items. Furthermore, the application offers another important feature, the polarity filter, the user having the opportunity to select articles with negative or positive polarity, depending on his/here option (Figure 8 b).



323 stiri in aplicatie
pentru institutia: Ministerul Educatiei Nationale
pentru domeniul: http://stiri.tvr.ro/

2020-01-06
Structura anului scolar 2020-2021 a fost lansata in dezbatere publica
http://stiri.tvr.ro/structura-anului-colar-2020-2021-a-fost-lansata-in-dezbatere-publica_854675_foto.html

Filtrul este aplicat pentru: data de start: 15/04/2016 si data de sfarsit: 15/04/2020 fiecare grafic 15 inregistrari

Figure 9: (a) Filter applied. (b) Wordcloud generated by the most used words.

To view the filtered news, the user should pass through options and finish the selection by pressing the button named “Aplica” (eng., Apply) located at the top right corner of the filter page (Figure 8 b). By pressing this button, a page with users’ interest articles is shown, giving more information about the selected items too, as shown in Figure 9 (a).

The user has another option in the right side of the filter page named “Statistici” (eng., Statistics). By clicking this section, a new page is opened with multiple statistical graphical representations for the selected data.

As illustrated in Figure 9 (b), the application generates a wordcloud image from the most used words of the day using python scripts, through application core model. Viewing this image, the user should understand the main topic of the day or the targets involved. Another image representing the number of news distributed on consecutive days is displayed in the same page giving to user the opportunity to browse through different periods.

To select a large time period the user must scroll vertically, loading and displaying more data in the same graphical representation.

Visualizing the total percent of negative, positive or neutral polarity for the daily news is another advantage of this application (Figure 10 a). Figure 10 (b) presents a bar chart containing the numbers of negative and positive news distributed on consecutive days representing a significant detail in discovering and analyzing those days with more negative news. Furthermore, from the polarity challenge perspective, the user has the opportunity to focus on the impact of negative messages spread in the virtual environment, by selecting and analyzing those online broadcasters with more critical news articles.

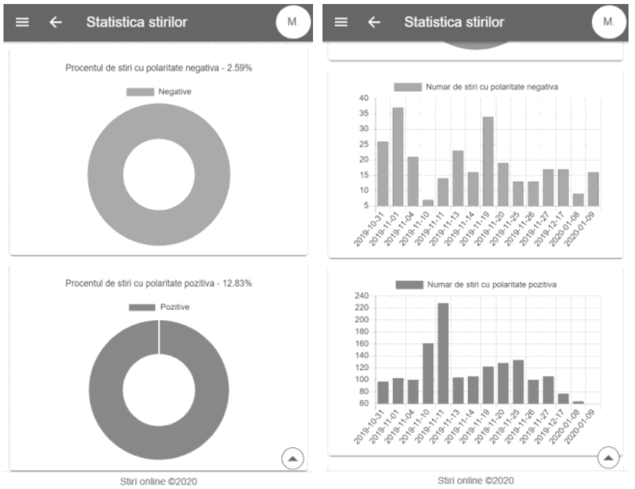


Figure 10: (a) Negative and positive polarities percent. (b) Number of news/day

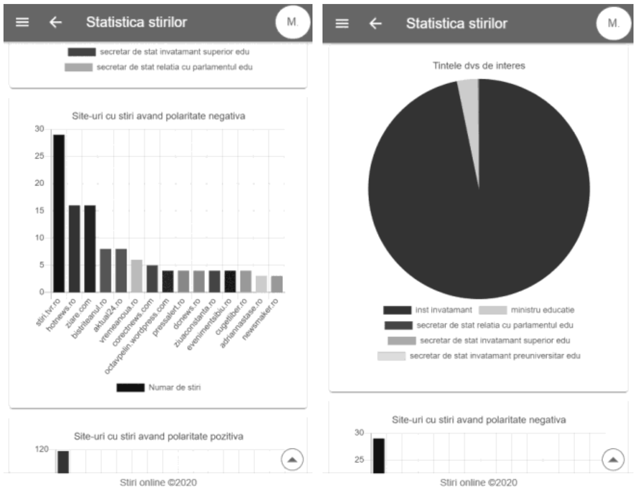

Statistica stirilor
M.
secretar de stat invatamant superior edu
secretar de stat relatia cu parlamentul edu
Site-uri cu stiri avand polaritate negativa
30
25
20
15
10
5
0
stiri.tvr.ro
hotnews.ro
ziare.com
bistriteanul.ro
aktual24.ro
vremeanoua.ro
corectnews.com
octavpelin.wordpress.com
pressalert.ro
dcnews.ro
ziuaconstanta.ro
evenimentsibiu.ro
cugetliber.ro
adriannastase.ro
newsmaker.ro
Numar de stiri
Site-uri cu stiri avand polaritate pozitiva
120
Tintele dvs de interes
inst invatamant
ministru educatie
secretar de stat relatia cu parlamentul edu
secretar de stat invatamant superior edu
secretar de stat invatamant preuniversitar edu
Site-uri cu stiri avand polaritate negativa
30
25
Stiri online ©2020

Figure 11: (a) Numbers of negative news/website. (b) Targets percents

Figure 11 (a) shows a chart describing the number of negative news, sorted in descendent order, for those online broadcasters whose most articles are with negative polarity.

As illustrated in Figure 11 (b), the application also generates a pie chart with the most used public targets. To discover these entities the crawler system applies a similar process with the one used by the management team in choosing keywords for companies/institutions.

Another aspect of the platform is represented by the different categories of news, such as international/national category or ironic/non ironic category, as shown in Figure 12 (a).

We tried to maintain similar features for the web and mobile application within the design process, but additional aspects of each application highlight the important differences. Each system has obvious advantages and disadvantages, such as speedup or small screens, but the most noticeable aspect of the mobile platform are the real time notifications. This module was implemented using One Signal notification library of the free JavaScript open source named Node.js and, beside Firebase Console we can push notifications to users with no cost, as shown in Figure 12 b.

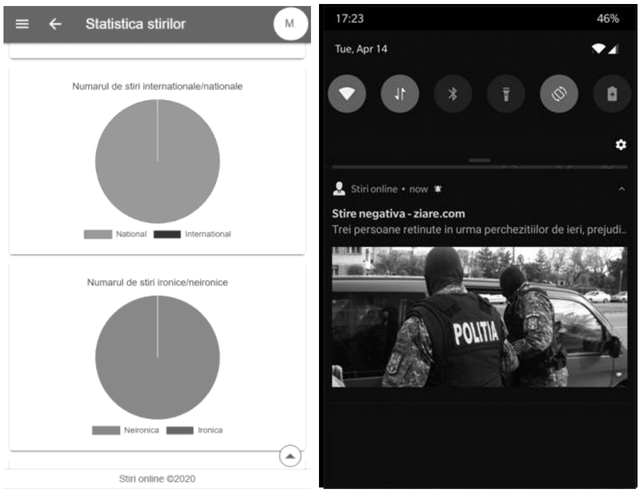

Statistica stirilor
M.
Numarul de stiri internationale/nationale
National
International
Numarul de stiri ironice/neironice
Neironica
Ironica
Stiri online ©2020
17:23
46%
Tue, Apr 14
Stiri online • now
Stire negativa - ziare.com
Trei persoane retinute in urma perchezitiilor de ieri, prejudi...
POLITIA

Figure 12: (a) News percent for different categories. (b) Push notification example

CONCLUSIONS

The most important online platforms, such as Facebook[3] and Google[4], announced in 2017 their commitment to use “trust indicators” to help users to select the reliable news of the online broadcasters and prevent spreading of the fake news. Even if these indicators were applied, this year, in the context of coronavirus pandemics, the misinformation circulate through social media applications, including closed stores, unproven methods for disease cure or military helicopters used to spray disinfectants against virus. Thus, the Romanian authorities had taken several measures in order to avoid spreading of false information, launching

[3] https://www.facebook.com/facebookmedia/blog/launching-new-trust-indicators-from-the-trust-project-for-news-on-facebook, last accessed on 5th July 2020

[4] https://www.blog.google/topics/journalism-news/sorting-through-information-help-trust-project/, last accessed on 5th July 2020

official online platforms[5] and closing other online domains[6].

Providing and sharing alternative explanations to fake news with logical meaning and scientific examples have been more effective than the above solutions, the main condition consisting of being immediately informed. Thus, the most important aspect of designing and developing this platform is that the registered user can receive notifications in real time about misinformation from the online environment regarding a public target or company/institution. Also, by having access to this application the user has the opportunity to analyze and visualize various statistical graphical representations.

Our main goal was to offer reliable content for readers and to increase the ability of the used models to detect and support registered users in their activities by receiving information about a specific target, company, institution or topic.

Another aspect of this application was to provide simple pages and well understood elements for displaying information in a structured way, helping to have an intuitive navigation and improve the user interaction and experience.

Nowadays, more online broadcasters are trying to use artificial intelligence approaches in the extraction and selection process of news, designing and developing crawler network.

There are several applications with Romanian content which present to users the latest news from the online environment for different topics, but with no archive or irony and fake detection features.

Evaluating an application is a complex task in the overall development process and we specially focus on creating the best user interface design and machine learning processes, but there is at least one indicator that can be reliable used in this platform, such as Firebase Console, which plays an important role for the future modules.

REFERENCES

1. E. Castledine, M. Wheeler, M. Eftos, "*Build Mobile Websites and Apps for Smart Devices*", USA, SitePoint, 2011.
2. P. Dominik, "*Mobile UI Design Patterns A Deeper Look At The Hottest Apps Today*", USA, UXPin, 2014.
3. T. McCann, "*The Art of the App Store - The Business of Apple Development*", Indianapolis, Indiana, USA, John Wiley & Sons, Inc., 2011.
4. R. Sandberg, M. Rollins, "*The Business of Android Apps Development - Making and Marketing Apps that Succeed on Google Play, Amazon App Store and More*", USA, Apress, 2013.
5. J. Iversen, M. Eierman, "*Learning Mobile App Development A Hands-on Guide to Building Apps with iOS and Android*", USA, Addison-Wesley Professional, 2013.
6. L. Sloan, A. Quan-Haase, "*The SAGE Handbook of Social Media Research Methods*", USA, Sage Publications Ltd, 2017.
7. M. Qiu, W. Dai, K. Gai, "*Mobile Applications development with android Technologies and Algorithms*", New York, USA, CRC Press, 2017.
8. J. Lumsden, "*Handbook of Research on User Interface Design and Evaluation for Mobile Technology*", New York, USA, Information Science Reference, 2008.
9. B. Bähr, "*Prototyping of User Interfaces for Mobile Applications*", Berlin, Germany, Springer, 2017.
10. E. Alepis, M. Virvou, "*Object-Oriented User Interfaces for Personalized Mobile Learning*", Warsaw, Poland, Springer, 2014.
11. C. Banga, J. Weinhold, "*Essential Mobile Interaction Design Perfecting Interface Design in Mobile Apps*", USA, Addison-Wesley Professional, 2014.
12. S. Hoober, E. Berkman, "*Designing Mobile Interfaces*", Canada, O'Reilly Media, 2011.
13. M.C. Buzea, Ş. Trăuşan-Matu, T. Rebedea, "*A Three Word-Level Approach Used in Machine Learning for Romanian Sentiment Analysis*", 18th RoEduNet, 2019.
14. M.C. Buzea, Ş. Trăuşan-Matu, T. Rebedea, "*Automatic Irony Detection for Romanian Online News*", 24th International Conference on System Theory, Control and Computing, ICSTCC 2020.

[5] https://datelazi.ro/ and https://stirioficiale.ro/, last accessed on 5th July 2020

[6] https://www.ancom.ro/en/decizii-decret-stare-de-urgenta_6253, last accessed on 5th July 2020

Analysis of the time spent on Facebook by Romanian university students

Iuliana Valentina Manea
Technical University of Civil Engineering
Splaiul Independentei 54, Bucharest, Romania
m.valentina@yahoo.ro

Costin Pribeanu
Academy of Romanian Scientists
Str. Ilfov No.3, Bucharest, Romania
costin.pribeanu@aosr.ro

DOI: 10.37789/rochi.2020.1.1.10

ABSTRACT

The use of Facebook by university students is a hot topic of research for more than ten years. However, few studies exist that are questioning the minutes spent daily for various tasks, such as socialization, posting/reading information, entertainment, and creation/update of the personal profile. Also, there are few studies investigating group differences as regards the various aspects of Facebook usage. This paper aims to analyze the differences by gender and year of study as regards the general characteristics of Facebook usage, motives for Facebook use, and time spent on Facebook for various activities. The results show that female students have larger Facebook networks and spend more time daily, especially for reading, posting, and sharing information. After the first year of study, students spend less time on entertainment and more time for editing the personal profile.

Keywords

Social networking websites, Facebook, university students.

ACM Classification

D.2.2: Design tools and techniques. H5.2 User interfaces.

INTRODUCTION

The popularity of Facebook among university students is continuously increasing. According to facebrands.ro, there were 9.6 million Romanian Facebook users by January 2017 out of which 21.47% were young users in the range 18-24 years. According to internetworldstats.com, the number of users in January 2020 was 10.86 million, 56.4% penetration rate.

University students are using social networking websites for different reasons: knowing new people, maintaining social relationships, collaboration, socialization, finding various information and resources, and entertainment [1, 3, 4, 6, 11, 12]. The daily use is spawned between a wide range of activities: chat, getting information and resources, finding out and sharing what is new, joining various groups of interest, promoting their image (personal profile), group work, and entertainment.

Although Facebook usage is an important research concern, few studies exist that are investigating the distribution of time spent daily for various tasks, such as socialization, posting/reading/sharing information, creation/update of the personal profile, and entertainment. Another issue is the analysis of group differences in Facebook usage. Since Facebook is widely used by young people starting from the high-school, another research question is which differences exist between the first-year students (freshmen) and other students.

This paper aims to analyze the distribution of time spent on Facebook. The time is analyzed separately by gender and year of study (first-year students vs. other students). Secondary goals are to analyze group differences as regards the general characteristics of usage and the motives for Facebook use. The study is using a data sample of 182 university students.

The rest of this paper is organized as follows. In the next section, related work is discussed. The method, data sample, and variables are presented in section 3. In section 4, three analyses are presented, as regards the general characteristics of Facebook usage, the motives for Facebook use, and the distribution of time spent daily for various tasks. The paper ends with a conclusion in section 5.

RELATED WORK

The time spent on Facebook by university students has an important contribution to their personal development by enhancing their communication skills and making them more sociable. Social networking websites have a positive influence on the bridging social capital of students [9] by enlarging social horizons, better communication with university people, an increased sense of belonging and participation [4, 12].

The studies of Selwin [10] and Ellison et al. [4] found that Facebook has a positive influence on students' formation, integration, and participation in the university community. Valenzuela et al. [12] found a positive correlation between the intensity of Facebook use and students' life satisfaction and civic engagement.

Junco [8] analyzed the relationship between the frequency of Facebook use, participation in Facebook activities, and student engagement on a large sample of 2368 college students. The results show that time spent on Facebook is positively related to co-curricular activities and real-world involvement on campus. In a similar vein, Selwin [S09] found that Facebook stimulates critical thinking and creates an open space for informal education of students.

The study of Cheung et al. [3] investigated the motives for Facebook use by students. They adopted the social influence theory, social presence theory, and the uses and gratification paradigm to explain why students are using Facebook and found that social presence was the most influential variable.

Yang and Bradford-Brown [11] analyzed the role played by social networking websites for the new students' adjustment to college. They found a positive correlation between social adjustment to college and using Facebook for maintaining social relationships. They also argued on the need to investigate specific activities rather than measuring the total time spent on social networking websites.

Chan et al. [2] explored the factors influencing user satisfaction in Facebook use and the moderating effect of gender on user satisfaction. They found that maintaining social relations and entertainment are the most important drivers of user satisfaction. While male students seemed to be more attracted by entertainment, female students were more interested in maintaining social relations.

Two previous studies explored general characteristics and motives for Facebook use on samples of Romanian university students. The former study [1], using a sample from a university of economics, found that the main motives for Facebook use were maintaining social relationships with high-school friends, socialization, and finding out what is new.

The latter study [6], using samples from two different universities, found that university students were using Facebook mainly for socialization, information, and collaboration purposes. An analysis of gender differences [5] showed that analyzed the time spent daily by university students from two universities and found that female students are more interested in maintaining social relations and less interested than male students in getting in touch with new people.

METHOD

The research questions are related to the differences by gender and the differences between first-year students and other studies as regards (1) general characteristics of Facebook usage, (2) motives for Facebook use, and (3) time spent on Facebook for various activities.

The data has been collected in May 2019 from a total of 182 students (127 males and 55 females) from the University of Building Engineering in Bucharest. The age of participants ranges between 18 and 34 years (M=20.36, SD=2.00).

After answering some several general questions as regards age, gender, and year of study they have been asked to answer questions related to the characteristics of usage, as shown in Table 1.

Table 1. Questions related to Facebook usage

How many FB friends do you have in your FB network?
How many of your FB friends are students in this university?
On average, how many days per week do you use Facebook?
On average, how many times per day do you log on Facebook? (1=once, 2=twice, 3=three and more, 4=continuous log)
On average, how many minutes per day do you use Facebook?

Next, students have been asked to answer six questions related to the distribution of time spent daily on Facebook, by indicating the percentage. The questions are presented in Table 2.

Table 2. Questions related to the distribution of time

Which is the weight of time spent on FB for socialization?
Which is the weight of time spent on FB for reading posts?
Which is the weight of time spent on FB for posting and sharing posts?
Which is the weight of time spent on FB for entertainment?
Which is the weight of time spent on FB for your Facebook profile?
Which is the weight of time spent on FB for other activities?

Then students have been asked to rate several statements on a 7-points Likert scale. The statements related to the motives for Facebook use are presented in Table 3.

Table 3. Motives for Facebook use

FBU1	I use Facebook to get in touch with new people
FBU2	I use Facebook to keep in touch with people I know
FBU3	I use Facebook to find information and resources
FBU4	I use Facebook for socialization purposes
FBU5	I use Facebook for collaboration purposes
FBU6	I use Facebook for entertainment purposes

Data analysis has been carried out using Lisrel 9.30 for Windows. The differences have been analyzed with one-way ANOVA.

ANALYSIS BY GENDER

General characteristics of Facebook use

The number of Facebook friends is large, varying from 6 to 5781 with a mean of 1173.13 (SD=1113.12). The number of Facebook friends from the university is varying from 2 to 500 with a mean of 61.17 (SD=72.65).

The frequency of use is measured by two variables: number of days/week and number of logs/day. The number of days/week is varying from 1 to 7 (M=5.54, SD=2.10). The number of logs/day is varying from 1 to 4 (M=2.72, SD=0.87). Most of the students (72.5%) are logging on Facebook two or three times per day. The time spent daily on Facebook is varying between 3 and 400 minutes (M=60.85, SD=68.69).

The characteristics of usage by gender are presented in Table 4.

Table 4. Characteristics of usage (N=182)

Variable	Male (N=127)	Female (N=55)
Facebook friends	1028.54	1507.00
Facebook friends students	45.76	96.75
Days / week	5.32	6.04
Logs / day	2.67	2.84
Minutes / day	54.61	75.27

All variables have higher mean values in the case of female students. A one-way ANOVA (1, 180, 181) showed that differences are statistically significant for Facebook friends (F=7.339, p=0.007), Facebook friends studying in the same university (F=20.994, p=0.000), and days/week (F=4.575, p=0.034). The difference is marginally significant for the minutes spent daily (F=3.522, p=0.062).

Motives for Facebook use

According to the mean values, the main motives for Facebook use are keeping in touch with known people, socialization, and entertainment. On the whole sample, only one item (getting in touch with new people) has a mean value below the neutral value (4.00 on a 7-points Likert scale).

Table 5. Motives for Facebook use (mean values)

Variable	Total		Male (N=127)	Female (N=55)
	M	SD		
FBU1	3.54	1.59	3.59	3.42
FBU2	5.69	1.55	5.44	6.25
FBU3	4.23	1.75	3.92	4.93
FBU4	5.18	1.66	5.04	5.51
FBU5	4.16	1.76	3.94	4.69
FBU6	5.14	1.82	5.04	5.36

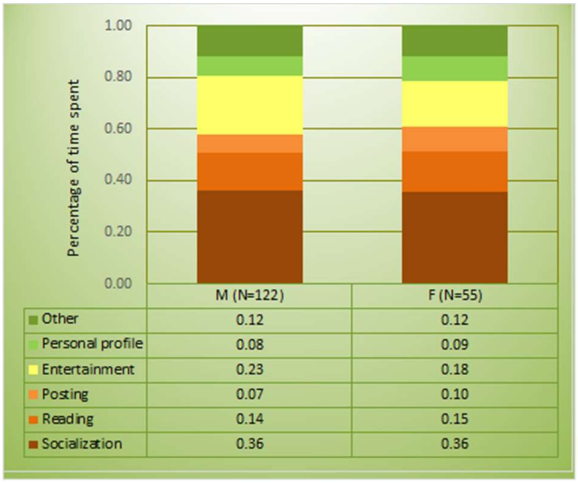
Comparison by gender shows large differences between the perception of male and female students. A one-way ANOVA (1, 180, 181) shows that these differences are statistically significant for FBU2 (F=11.215, p=0.001), FBU3 (F=13.631, p=0.000), and FBU5 (F=7.308, p=0.008) and marginally significant for FBU4 (F=3.120, p=0.079).

Overall, female students rated higher all items, except for the first (getting in touch with new people).

Distribution of time spent on Facebook

For the whole sample, the most time-consuming task is socialization (chat) which accounts for 36% (SD=0.26) from the total time. The next time slots are spent for entertainment (M=0.21, SD=0.08), reading information (M=0.15, SD=0.12), posting and sharing information (M=0.08, SD=0.08), and updating the personal profile (M=0.08, SD=0.08). Other activities account for 12% of the time (SD=0.11)

The differences by gender as regards the distribution of time spent on Facebook is presented in Figure 2.



	M (N=122)	F (N=55)
Other	0.12	0.12
Personal profile	0.08	0.09
Entertainment	0.23	0.18
Posting	0.07	0.10
Reading	0.14	0.15
Socialization	0.36	0.36

Figure 1. Distribution of time spent on Facebook

Male students are spending more time than female students for entertainment and less time for updating the personal, reading, and posting/sharing information. However, a one-way ANOVA (1, 180, 181) shows that differences are marginally significant only for posting /sharing information (F=3.17, p=0.077).

ANALYSIS BY YEAR OF STUDY

General characteristics of Facebook use

The gender differences as regards the characteristics of usage are presented in Table 6 (mean values).

.Table 6. Characteristics of usage (N=182)

Variable	1st-year (N=122)	Other (N=60)
Facebook friends	1096.16	1329.63
Facebook friends students	47.95	88.97
Days / week	5.47	5.67
Logs / day	2.73	2.72
Minutes / day	59.79	63.02

Almost all variables have lower mean values for the first-year students. The number of Facebook friends that are students in the same university is much lower. A one-way ANOVA (1, 180, 181) showed that only this difference is statistically significant (F=13.84, p=0.000).

Motives for Facebook use

The differences as regards the motives for Facebook use are presented in Table 7 (mean values and standard deviation on total, mean values by year of study).

Table 7. Motives for Facebook use (mean values)

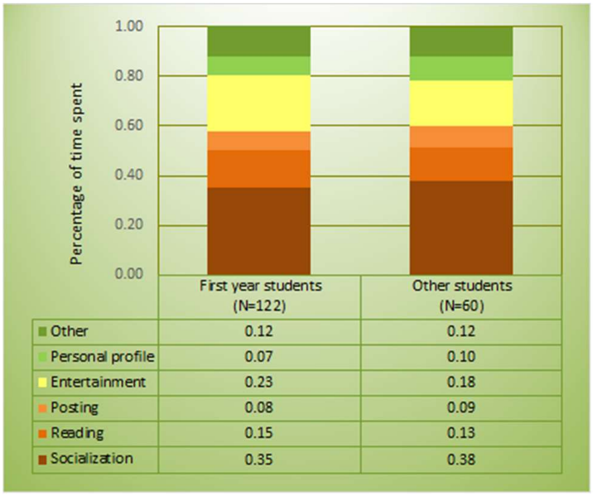
Variable	Total		1st-year (N=122)	Other (N=60)
	M	SD		
FBU1	3.54	1.59	3.51	3.60
FBU2	5.69	1.55	5.69	5.68
FBU3	4.23	1.75	4.17	4.33
FBU4	5.18	1.66	5.18	5.18
FBU5	4.16	1.76	3.98	4.53
FBU6	5.14	1.82	5.13	5.15

A comparison between 1st-year students and other students shows similar mean values for three motives (FBU2, FBU4, and FBU5) and the same order of preferences. For the other three reasons, the mean values are lower for 1st-year students. A one-way ANOVA (1, 180, 181) shows that the difference is statistically significant for FBU5 (F=4.00, p=0.05) which suggests that after the first year of study students are more interested to use Facebook for collaboration purposes.

Distribution of time spent on Facebook

The distribution of time spent on Facebook for various activities is different for 1st-year students, as shown in Figure 1 (mean values).

Freshmen are spending more time than other students for reading information and entertainment and less time for socialization, updating the personal profile, and posting /sharing information. This finding supports the idea that Facebook may play a positive role in students' adjustment to college. A one-way ANOVA (1, 180, 181) shows that differences are marginally significant only for updating the personal profile (F=3.574, p=0.060).



	First year students (N=122)	Other students (N=60)
🟩 Other	0.12	0.12
🟩 Personal profile	0.07	0.10
🟨 Entertainment	0.23	0.18
🟧 Posting	0.08	0.09
🟧 Reading	0.15	0.13
🟫 Socialization	0.35	0.38

Figure 2. Distribution of time spent on Facebook

Discussion

The results of this study show that Romanian university students have a large Facebook network and spend a lot of time on Facebook. The main reasons for Facebook use are keeping in touch with known people (maintaining social relations), socialization, and entertainment. In this respect, the findings are similar to the results of other studies [3, 11], as well as with the results of previous studies in Romania [1, 6].

The mean values of the variable measuring the motives for Facebook use are consistent with the distribution of time spent for these reasons. Socialization and entertainment account for more than 50% from the total time spent daily on Facebook by university students. As regards the time spent on entertainment, the findings confirm the previous results [7] showing higher mean values for first-year students.

As regards the gender differences, the findings confirm the results of other studies [5, 6]. Male students are more interested in entertainment than their female colleagues. Female students spend more time than male students for reading, posting, and sharing information on Facebook.

There are limitations of this exploratory study. First, given the cross-sectional nature of the study, it is not possible to measure the evolution in time of students' behavior. Second, the sample is relatively small and the number of 1st-year students is twice the number of other students. Last, but not least, as many authors pointed out [11, 12], the motives for Facebook use and the actual usage depend on a diversity of factors. Therefore, the conclusions of this study using a sample of college students from a particular university should not be generalized to other populations.

CONCLUSION AND FUTURE WORK

This study contributes to a better understanding of the time spent on Facebook by Romanian university students. The results show that socialization and entertainment are the main reasons why students are using Facebook and account together for more than half of the time spent daily on this social networking website.

Future work should continue in two research directions. The first direction is to enlarge the number of motives for Facebook use and refine the set of activities for the actual use. The second should be a longitudinal study to assess the evolution of time of Facebook-related behavior.

REFERENCES

1. Balog, A., Pribeanu, C. Ivan, I. (2015) Motives and characteristics of Facebook use by students from a Romanian university. In: Dardala, M., Rebedea, T.E. (Eds.) *Proceedings of RoCHI 2015*, Bucharest, 24-25 September, 137-140.
2. Chan, T. K., Cheung, C. M., Shi, N., & Lee, M. K. (2015). Gender differences in satisfaction with Facebook users. *Industrial Management & Data Systems*, 115(1), 182-206, https://doi.org/10.1108/IMDS-08-2014-0234
3. Cheung, K., Chiu, P.-Y., Lee, M. (2011) Online social networks: Why do students use Facebook. *Computers in Human Behavior*, 27(4), 1337-1343. https://doi.org/10.1016/j.chb.2010.07.028
4. Ellison, N.B., Steinfield, C., Lampe, C. (2007). The benefits of Facebook "Friends:" Social capital and college students' use of online social network sites, *Journal of Computer-Mediated Communication 12*, 1143-1168.
5. Iordache, D. D. (2017). Gender differences in the motives of using online social networks by university students. *Proc. eLSE 2018*, Vol. 2, 570-577. DOI:10.12753/2066-026X-17-166
6. Iordache, D. D., & Pribeanu, C. (2016). Exploring the motives of using Facebook – a multidimensional approach. *Revista Romana de Interactiune Om-Calculator*, 9(1), 19-34.
7. Iordache DD, Pribeanu C, Gorghiu G, Manea VI (2018) Distribution of time spent on Facebook by students from two Romanian universities. *Proc. of IE 2018 Conference,* Iasi 17-18 May, 195-200.
8. Junco, R. (2012). The relationship between frequency of Facebook use, participation in Facebook activities, and student engagement. *Computers & Education*, *58*(1), 162-171.
9. Putnam, R. D. (2000). *Bowling Alone*. New York: Simon & Schuster.
10. Selwyn, N. (2009). Faceworking: exploring students' education-related use of Facebook. *Learning, Media, and Technology 34(2)*, 157-174.
11. Yang, C. C., & Brown, B. B. (2013). Motives for using Facebook, patterns of Facebook activities, and late adolescents' social adjustment to college. *Journal of youth and adolescence*, 42(3), 403-416. https://doi.org/10.1007/s10964-012-9836-x
12. Valenzuela, S., Park, N., & Kee, K. F. (2009). Is There Social Capital in a Social Network Site?: Facebook Use and College Students' Life Satisfaction, Trust, and Participation. *Journal of Computer-Mediated Communication*, *14*(4), 875-901.

JIST: Java Interaction Separation Toolkit

Adrian-Radu Macocian
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
rmacocian@gmail.com

Dorian Gorgan
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
dorian.gorgan@cs.utcluj.ro

DOI: 10.37789/rochi.2020.1.1.11

ABSTRACT

A GUI toolkit is a library consisting of the elements needed for developing interactive applications. In the 2000s, a lot of effort was devoted to building platforms that enabled the creation of rich internet applications. This let the field of desktop applications underdeveloped. The Java Interaction Separation Toolkit (JIST) was developed with the intention of having a lightweight, cross-platform and support for a declarative UI. By using the WORA aspect of Java most of the desktop platforms are covered. It features XML support for describing user interfaces in a more natural way than the classic wall of text associated with the native code-behind approach.

Author Keywords

Java; GUI sub-system; Graphical User Interface; Interactive Applications; Markup Language; Functional and Interaction Separation; GUI toolkit.

ACM Classification Keywords

H.5. Information interfaces and presentation (e.g., HCI)

INTRODUCTION

Graphical User Interfaces are the main reason why the personal computer reached the mainstream success it has now. In year 1960 the idea of a Graphic User Interface (GUI) started to take shape, and it changed a few times until it reached its peak together with the historical launch of the Window 95 in 1995 [2]. Most current GUI toolkits were created in the late 90s, early 00s. They may be well maintained, but the foundations of the toolkits were created in a time when the environment for graphical interfaces was completely different, and that takes its toll. Computation power is abundant and it’s becoming easier to provide the functional requirements of an application. In these circumstances, the choice of software is made based on user friendliness and fluidity of the interface design.

Developing user interfaces is mostly done using a GUI toolkit or framework. Those toolkits handle all the hardware inputs and outputs, define the interaction techniques and provide the developer with the tools for giving input choices to the user, and for handling that input from the user.

This paper will describe the Java Interaction Separation Toolkit (JIST). It is a GUI toolkit developed completely in Java with no external dependencies, that focuses on separating the functional and the interactive components of interactive applications using a markup language. By avoiding any external dependencies, it is ensured that the toolkit may be used on any system that supports the Java Virtual Machine (JVM).

MOTIVATION

It is possible to create a complex user interface using the existing solutions, but it is unnecessary difficult. The current solutions (especially in Java) require nesting of elements using layout managers to ensure that the software will look the same independently of the platform on which it runs. This causes walls of text and makes it so that the code is impossible to be read.

Markup languages can be used to provide layouts to elements in a more natural way and makes visualizing those layouts easier. Windows WPF takes full advantage of this aspect with the XAML description of interfaces [8][9].

There was an attempt of having markup language support in Java with the JavaFX and the FXML (an XML-based language used for describing user interfaces), but JavaFX’s future is an uncertainty at this point [7]. Even without the uncertainty surrounding JavaFX’s future, using FXML is difficult and seems like an additional feature instead of a core functionality of the system.

The purpose of JIST is to create a platform for describing user interfaces which can separate the aspect from the behavior of the application. This decoupling can enable teams to work concurrently and makes the software easier to understand and maintain. The problem of having software look the same, independently of the platform can be solved by providing context-relative sizes and locations. By specifying everything relative to another element, the developer should always understand how the application should look. This way it is possible to achieve similar displays independent of the system which runs them, without the need of using multiple nested layout managers.

The combination of providing a separation between the functional and the interactive component, providing contextual sizes and location to elements, and the ease of developing layouts in xml is the reason why JIST brings a new approach to the field of GUI toolkits.

OBJECTIVES

The main objective of this paper is having a cross-platform solution for developing Graphical User Interfaces which supports a markup language for the description of the layouts of applications.

Cross-Platform

The platform most often represents the operating system which the application runs on. Covering more than 97% of the market share of desktop OS is done by making sure that the system can be ran in Windows, OS X and Linux [4]. Since all the aforementioned operating systems are able to run a Java Virtual Machine (JVM), developing JIST in pure java should be able to cover the cross-platform objective.

Figure 1 presents the desktop operating system market share as measured by [4].

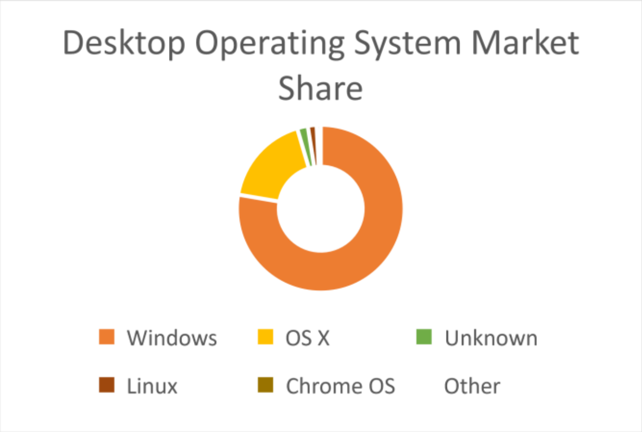


Figure 1. Desktop Operating Systems Market Share [4]

Contextual Size and Location

Using absolute sizes and locations is, rightly so, frowned upon when developing user interfaces. That is because it is impossible to know on what screen the software will be run on, so therefore we can’t predict how the software will look like. Current Java native solutions have solved this issue with the use of layout managers. The problem for a complex user application, there will need to be a lot of nested layout managers, which makes it hard to keep track of everything.

The solution used in JIST is one that is already present in the field of web applications, more specifically in HTML. That is, using locations and sizes relative to the element on which the component is displayed. This way, the size of the screen should not influence the overall look of the user interfaces.

Markup Language Support

The separation between the functional and the interactive components of an interactive application could be achieved, in the way JavaFX [3] and WPF [1] also achieved this, by allowing the interface to be described in a markup language. Most GUI toolkits store the elements into a tree-like structure, which is conceptualized more easily in a markup language format.

The support for a markup language was considered from the very beginning of JIST, which ensures that all the components are designed with the goal of supporting markup language in mind. It is important that the markup language feels as a part of the framework, not some feature that may or may not be complete.

The support for the markup language also helps with the problem of having multiple nested layouts. This is due to the nature of markup languages, which allows the visualization tree-like structures in a natural way, as opposed to normal programming languages where it is almost impossible to visualize multiple levels of nested layouts.

ANALISYS

In this section the theoretical foundation on which the project was created will be provided. Here the paper will go a little more in depth into interactive applications, since it is important to know how a tool needs to be used, before designing the tool.

Interaction Applications

An interactive application is composed of two big, and ideally separate, components. The Functional Component, where all the abstract operations on objects are happening and the Interactive Component where the interaction techniques are described together with the interface of objects and operations on those interfaces. The user can only see and act upon the interactive component. The interactive component takes all the input from the user and first validates it, and then processes and transforms it into an application operation that is passed to the functional component.

Interaction techniques are the way in which the user, with the help of the hardware resources and given software components, may provide information to the computer. The results of the interaction are usually visible on screen.

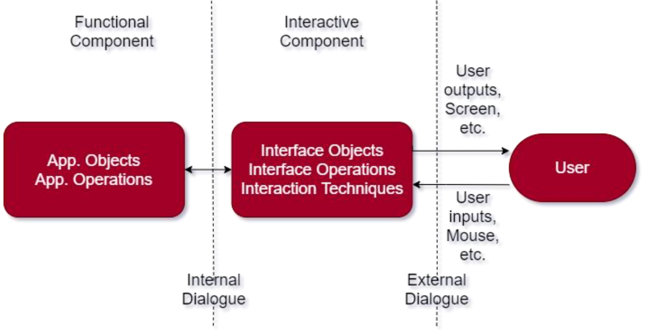


Figure 2. Structure of an Interactive Application [5]

Interaction Techniques

An interaction technique may be informally described as an element which can be graphically represented on screen. Formally, an interaction technique is a way in which the user, with the help of the hardware resources and software components, may provide information to the computer. The interaction technique usually is composed of an input device and an interaction element.

Model of an Interaction Technique

An interaction technique is the way in which a user may communicate with an application with the purpose of achieving a simple action. It may be a simpler way of visualizing the actual communication. Most interaction techniques can be described in the format of an interaction cycle.

The interaction cycle is composed of a prompter, symbol, echo and value. This interaction cycle helps with the better visualization of how a user communicates with the software. An Interaction technique begins with the prompter stage of the cycle, when something is selected or focalized, and the system lets the user know that some form of input is accepted. In the symbol stage, the user will provide some input which will be validated. The echo is the system's way to show some feedback to the user to confirm that the input was received and, finally, in the value stage, the value will be modified to what the application accepts (i.e. normalization).

In most techniques, the user has multiple possible available valid actions (such as clicking, dragging, moving the mouse). An interaction technique can be a metaphor or a symbolical representation of a real operation, which should help us visualize the operations. The metaphor has a visual presentation (some shape or drawing on the screen), a scenario (a way in the user may interact with it), a sequence of user actions (a set of permitted actions) and an interaction device (usually an input device: mouse, keyboard, etc.).

Event Based Control

Most interactive applications are event driven. This means that during most of its lifecycle, the application is waiting for some user events to happen, to which it will respond based on some predefined procedures. The response time to those events must be as low as possible for a satisfying experience for the user.

Figure 3 shows the flow of an event-based control.

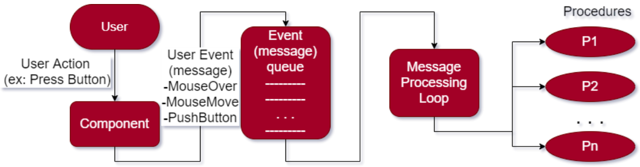


Figure 3. The structure of event-based control [6]

In the event-based control of the application, the user first performs an action which an interaction technique will capture. That action is then transformed into an event that is passed to an event queue. The event queue works as a FIFO 1 list. The queue passes the events, in order, to a message processing loop. The loop decides on which procedure the call for each message. A procedure is a set of actions that are designed to resolve the interaction with the user.

Conceptual Architecture

We will start discussing the conceptual architecture with the most basic objective: Displaying objects on the screen. In order to display something on the screen we need to use OS system calls for creating a new window and then for drawing on that window. A specialized library for hardware interaction will be used. This will also solve the problem of receiving and interpreting user input.

Creating the aspect of the application can be represented as a tree of graphical elements, where the leaves are in the front of the screen and the root in the back of the screen. In order to create such a tree, a common class is needed, which will act as the nodes in the tree. This class should also implement all the methods needed by most of the visual components (such as painting, checking for collision, setting the location and size, etc.). This tree of elements should be passed to the window and then the window will display them on the screen.

It was noted that supporting a markup language for the layouts is a big objective. The markup language should be interpreted at run-time and then a visual tree should be generated following a description in markup language. The choice to use the standard XML notation for the layouts was done due to its flexibility and structure [5].

Besides the visual elements, for modularity, there should be an extra element that deals with the decorations (such as borders and effects). Having them described separately from the main class will give more flexibility in designing applications and for future changes.

With all those choices in mind, the next step is to present a conceptual architecture for the system (Figure 4). Two libraries have been added which are present in the native JDK so that any system that is compatible with Java will be compatible with this system. Those two libraries are: Swing for interacting with the hardware, and XML DOM which is used by the parser.

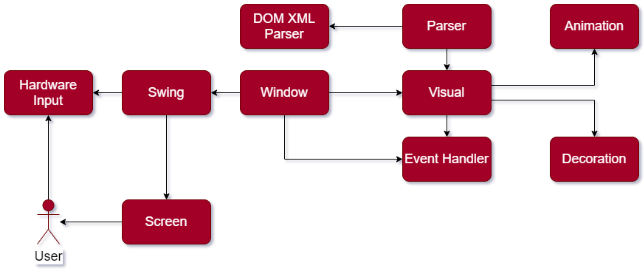


Figure 4. The conceptual architecture of JIST

RELATED WORK

With user interfaces being as big as they currently are, naturally there exist a lot of current frameworks for developing GUIs. It is not practical to try to compare this solution to all of the other available solutions, as they are so numerous. There also isn't a universal best toolkit in this field, and everyone has their own preferences. The following paragraphs will describe two of the most widely used frameworks in Swing and WPF. JavaFX will also be presented as at one point it was supposed to be the successor of Swing.

Swing

Swing was designed with a modified model-view-controller design pattern. It uses the UI component as both the view and the controller. It was developed entirely in JAVA for cross-platform support and easier maintenance. It supports multiple look-and-feels so that it feels native in the platform it runs in. But the look-and-feel of the application may also be changed at runtime.

Swing was developed as an upgrade to the existent AWT API, so it has full compatibility with AWT components. It handles look-and-feel characteristics in a UIManager class, which communicates with each component's UI object to control the display.

JavaFX

It was initially released in 2008 as the successor to Swing, which was supposed to create both web and desktop applications with ease. Since then the web application support has been deprecated and JavaFX started focusing solely on desktop applications. It features its own markup language, the FXML, for declarative description of interfaces. The structure is separated into stages and scenes. Each stage is a window, but it may support multiple scenes, although only one scene is active at a time. All the elements in a scene create a scene graph. The user interface is not native, but it supports Cascading Style Sheets (CSS) for personal touches to applications.

Windows Presentation Foundation (WPF)

WPF: Windows Presentation Foundation is the graphical sub-system developed by .Net Foundation under Microsoft. It was released in open source in December 2018 together with WinForm and WinUI and is the go-to system for developing Graphical User Interfaces using the .Net framework.

All display in WPF is done through DirectX so it relies on Windows for it to function. This also means that it is significantly more efficient in hardware and software rendering. It is usually the go-to platform for developing desktop applications that are only supposed to work on Windows.

WPF values properties a lot higher than events. The goal is for the system to have multiple properties that control the flow of the application. Changes are signaled through notifications. Dependencies are handled automatically, and any property change triggers a dependency revalidation. Any object can provide other objects definitions of its properties.

IMPLEMENTATION

Here the design choices and how most of the framework was implemented will be laid out.

Storing Elements in Memory

The Window class has an instance of a Java Swing JFrame which deals with drawing the final virtual image on the screen. The reason for using Swing is that this ensures the platform has as few dependencies as possible, and the Swing library is included in the native JDK. Besides this, Swing handles all the system calls for hardware interrupts. The window class acts as an interface between this solution and the Swing library.

The Visual class is the backbone of the entire structure. Through this class all the information that should not be accessible to the user is shared, such as the virtual images of the components and handling of user events. With the help of this class, the system may create a visual tree which will later be used for passing graphics information (figure 5). Each node in the tree (which is visible on screen) has a virtual image assigned.

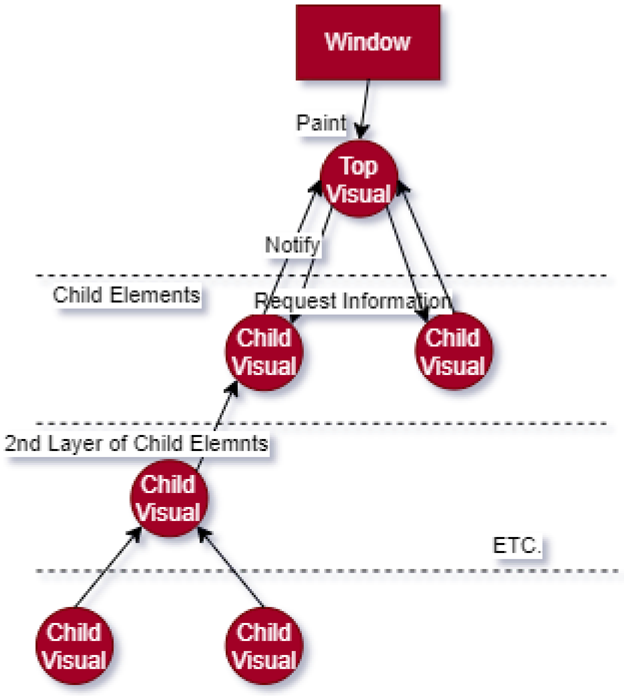


Figure 5. Conceptualization of the visual tree

Any parent node has access to all its children nodes through the findByName method. The parent node decides when and how to place the children nodes in its visual image. A node that sustained a change which requires a repaint needs to signal all the way up the tree that the repaint is necessary. The request is propagated up the tree and only the direct ancestors of that node will need to be repainted while the other nodes remain valid.

The only way for children to pass information to a parent is through notifications, which the parent decides if and how to handle.

Event Handling

The importance of user events for any graphical user interface cannot be overstated, so triggering them appropriately is a must. There are some rules which describe how events are passed to the components on the screen:

- Only one element on the screen can have the focus
- The mouse is considered on top of an element if the collision method (isInside) for the mouse coordinates returns true
- Mouse events are triggered only on the front-most component (which returned true for isInside)

The framework currently supports three kinds of events: mouse events, keyboard events and mouse wheel events.

The window gets the hardware information from Swing and then passes that information down the visual tree until the right component is reached. The keyboard, mouse pressed, and mouse wheel events are passed directly to the focused element, which is stored as a singleton in the window.

Drawing on the Screen

Each visual component is assigned a virtual image the moment when it is added to a window (so it is displayable). The window class extends the visual class, so it also has a virtual image, which is passed to the JFrame the moment a frame needs to be drawn. This ensures that all the frames drawn on screen are complete images using double buffering.

Every node first applies its own graphic logic on the virtual image and afterwards paints the child nodes' virtual images on top of its own image. This way, if any node in the visual tree needs to be updated, it will only affect the direct ancestor nodes. Any other node can keep painting the same virtual image with no repainting needed.

The moment a new window is created, a *Painting Thread* will also be created. The window also stores the information on the number of frames to be displayed per second. The painting thread makes sure that frames are displayed at the correct rate, and that each image is the most up to date image the system has.

The painting algorithm is composed of two methods: revalidate and repaint. The thread calls the repaint method for every frame. The method first runs all the animations, and then the method checks if any component needs revalidation, if this is not the case, then the virtual image of the Window is still up to date and can be displayed on the screen as is. If a component was changed and the image needs to be updated, then the revalidation method is called. In revalidation, any outdated nodes in the visual tree will clear their virtual images, and then proceed to repaint them to be up to date.

Since when a node requests an update, all the ancestors of that given node need to be updated also, all the updates requests will propagate all the way up to the window. This way, if the window doesn't require any updates, neither does any other node in the visual tree, and the check can be done in O(1).

Window

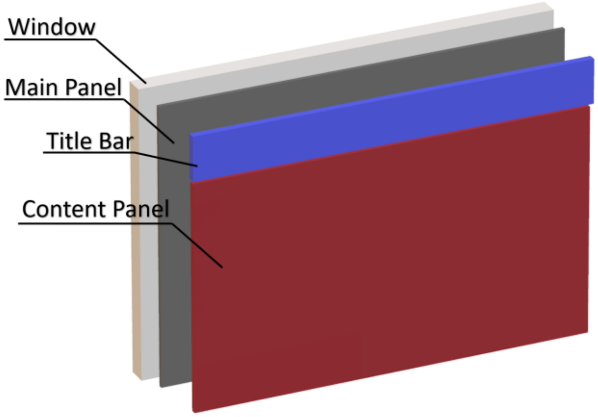


Figure 6. The structure of the window

Any GUI application using JIST requires a Window to function. All the visual elements must be placed inside a visual tree which has a window as the root. This ensures both that the element is displayed on screen, and that the user can interact with that element using the mouse or the keyboard.

The window is first created with a generic title bar. The title bar may be removed or replaced at any time. Any new title bar must extend the title bar class. The active title bar is also an instance of Visual so some attributes such as the colors may be changed as necessary without the need to change the entire title bar.

The window is composed of a Main Panel which stores a Content Panel and a Title Bar (Figure 6). Any component added with the add Visual method to the window, is automatically added to the Content Panel.

Animations

Animations are a way of displaying multiple images in a very short period which gives us the feeling of movement. Small animations can give an, otherwise bland, application a livelier feel.

Since the animations need to change with each frame, triggering the animations is done in the repaint method, triggered by the Painting Thread. This ensures that before each virtual image is validated, each existing animation is executed with one step.

To make sure that developers can implement their animations with as little hustle as possible, an Animation

Interface is created, which specifies the number of frames per second and a step method, which returns false while the animation is running and true when the animation is done. This is done so that the painting thread can remove finished animations from the window's list of animations.

Animations can be added on any node in the visual tree, and they will be recursively passed all the way up to the window, which stores a list of all the running animations from the components displayed in it.

There are currently 3 types of animations that are used by existing components: color, location and size animations.

XML Parser

Using the XML as the declarative markup language, makes it possible for the system to avoid any external dependencies and to keep the system lightweight.

The parser uses reflection to search for all the classes in the current project or .jar executable, and then matches the tags in xml to classes. This makes it possible to just create a new class which will be usable in xml description right away. The only condition for a class to be declarable in xml is that the class must extend, directly or indirectly, the visual class.

The only knowledge that the parser requires is a string to the xml file that is going to be parsed.

The root element from the XML (which usually is the Window) is first instantiated and has its attributes set, the same goes for all the child nodes until the file is covered. After all the instances are created and have had their attributes set, the parser starts returning bottom-up adding all the nodes to their parent nodes.

Hardware Acceleration

Although it was not an initial requirement, it was important to give to the developers the option of enabling hardware acceleration for the drawing. The first step in enabling hardware acceleration in Java is to set the flags in the JVM. The flags must be set before any graphical processing is done, so it is important that hardware acceleration is enabled first in the project if needed. The second step is setting a flag in the visual class which will cause all the virtual images created to be changed from bufferedImage to volatileImage to ensure that the entire advantage of the hardware acceleration is used.

Contextual Size and Location

The location of an element is given by a locationPlacer, which receives the size of the element and of its parent element, and then it decides on where the element should be placed. The placers are created through a factory pattern so that developers can create their own placing logic. As of right now, there are 10 available placers: top-right, top-center, top-left, middle-right, middle-center, middle-left, bottom-right, bottom-center, bottom-left and a general placer. The first 9 placers do exactly what their name suggest.

The general placer has two parameters, a relative position and an absolute position. The relative position is given in the form of two float numbers between 0 and 1 and describes the position inside the parent element. If the relative position is missing, then the placer will use the absolute position for the location of the element.

The size of elements is decided similarly to how the general placer chooses the location of elements.

EXPERIMENTAL EVALUATION

There were three types of testing done for JIST. Performance testing, scalability testing and integration testing. Afterwards, an evaluation for the usability of the system is provided.

Performance Testing

The performance of the system is decided by the rate in which frames may be repainted, while increasing the depth of the visual tree. The test consists in creating a new window of size 1024 x 576 and adding a panel of the same size. Then before every repaint, ask for the panel to be revalidated and save the number of frames displayed on the screen in one second. To avoid erroneous data, the test was repeated 60 times. After that, a new panel of the same size was added as a child to the last panel, thus deepening the visual tree. Now the new panel was asked for revalidation, which would cause both the panels and the window to be revalidated. The same pattern was repeated until a depth of 30 elements in the visual tree was reached.

The entire test was done two times, the first time the system had no hardware acceleration, and the second time hardware acceleration was activated.

The test checks the performance in the case of constant revalidations which is usually seen in games. Static applications don't usually need to revalidate the image before each frame, but even in these circumstances, without the use of hardware acceleration JIST can display over 30 FPS up to a depth of 7 nested elements (Figure 7).

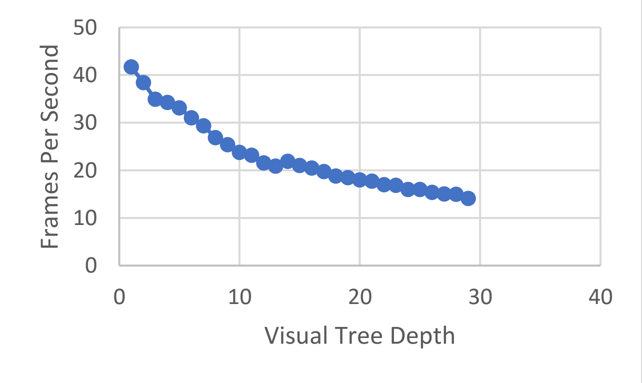


Figure 7. The FPS graph without hardware acceleration

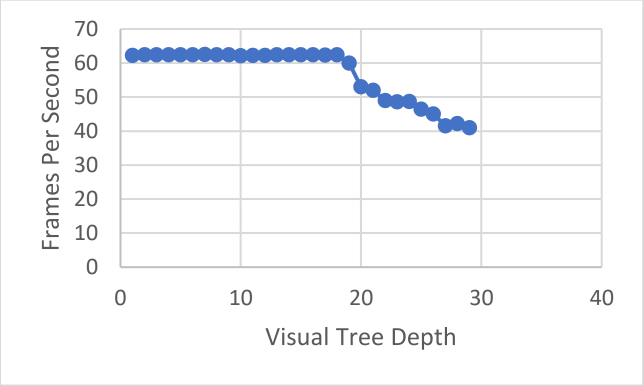


Figure 8. The FPS graph with hardware acceleration

When using hardware acceleration JIST could display 60 FPS, which was the capped value, up until the depth of 17. Even on the depth of 30 the system could display at a rate of 40 FPS.

Scalability Testing

The scalability of JIST may be tested, by checking the number of components that can be added on a visual tree. Again, two cases were considered. The first case was adding elements with a size of 0x0. The test was stopped after 10,000 elements were added, because the system showed no signs of slowing down or troubles.

The second test was done by adding elements of the same size as the window (1024 x 576). This time the creation of elements took a longer time and it seemed as the system would crash. The problem is that each component is given a virtual image of its own size, and the system runs the risk of running out of memory. But in our test case the JVM would always be able to allocate more memory before the system would run out of memory. The test was stopped at 3500 elements, but after 1000 element the creation of new elements started to take considerably longer.

It is worth noting that the elements were all added at the same depth inside the visual tree, to avoid any recursive calls and stack overflow errors. It is very hard to imagine a real case scenario where a user might need more than 3500 elements the size of the screen. The test does show that it takes considerably more time to create new elements the larger they are, and this causes rises in the response time and falls in performance while the system handles the creation of the element. The response time and performance quickly readjust once the elements are created.

Integration

For integration testing, multiple applications were developed. All through the development of JIST new applications were developed with the purpose of seeing how the system handles real scenarios.

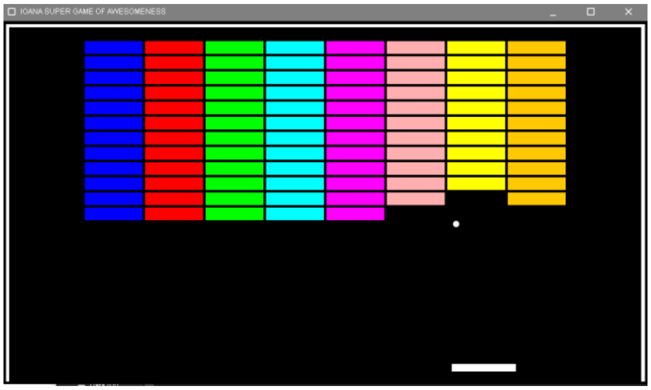


Figure 9. Breakout replica

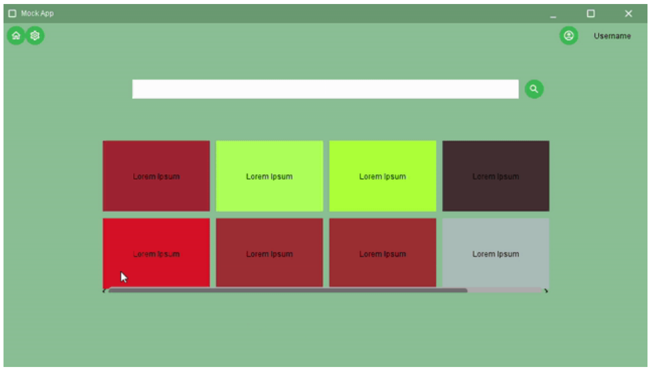


Figure 10. A mock application

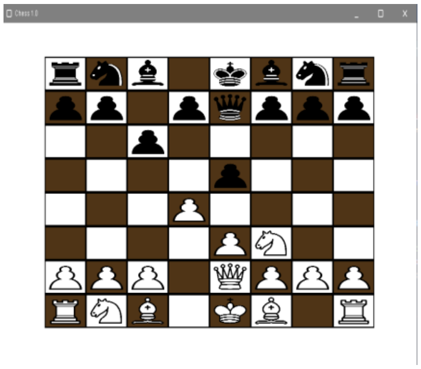


Figure 11. A Chess game

All the above figures (figures 9, 10 and 11) are applications developed completely in JIST. The layouts were written completely in xml.

Usability

It is hard to rate the usability of such a project objectively, since the toolkit choice of each developer is very much subjective.

The usability is described in [6] as: “the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”. The usability is highly related to the target audience of the system. The target audience is the software developers that want to create desktop graphical user interfaces in Java.

Several efforts were made to make the toolkit as friendly as possible to new developers and to keep it extendable so that everyone may implement their own vision. A few examples of those efforts are making sure that any class is usable in xml description, making sure that every functionality can be extended upon and providing a set of interaction techniques or widgets that are required in almost any interface. The set of available components is still expanding, but currently consists of:

- Buttons and toggle button
- Text boxes and input text boxes
- Panels, scrollable panels and grid panels
- Check boxes and radio buttons
- Dropdown menus
- Sliders
- Images

Another part of usability was giving the developers the possibility to reference images by just specifying the name. The developer can just add an image in .png format to the class-path and reference it through just the name.

CONCLUSIONS

In conclusion, it is entirely possible to develop both static applications and games using JIST. The final library is lightweight and with a set of icons bundled into it, the size does not exceed 200 KB.

What separates JIST from other available solutions is that: it was developed with the intention of describing layouts in a markup language, it is lightweight, and it is easy and straightforward to use without compromising in the performance, customizability or response time departments.

REFERENCES

1. Anderson, C. "*Essential Windows Presentation Foundation (WPF)*", Addison-Wesley Professional, 2009
2. Barnes, S. B. "User friendly: A short history of the graphical user interface", *Sacred Heart University Review: Vol 16*, *Issue 1*, Article 4, 2010
3. Clark, J., Connors, J., Bruno, E. "*JavaFX: Developing Rich Internet Applications*", Addison-Wesley Professional, 2009
4. Desktop Operating System Market Share Worldwide, https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-201906-202006, visited: 20-jun-2020
5. Harold, E. R. "*Processing XML with JavaTM: A Guide to SAX, DOM, JDOM, JAXP, and TrAX* ", Addison-Wesley Professional, 2002
6. IOS, "*Ergonomics of human-system interaction — part 11: Usability: Definitions and concepts*", 2018
7. Oracle, "*Java Client Roadmap Update*", 2018, https://www.oracle.com/technetwork/java/javase/javaclientroadmapupdate2018mar-4414431.pdf, visited: 17-nov-2019
8. Subhashini, C., Premalatha, S., "XAML - a user interface markup language", i-*manager's Journal on Software Engineering, 4(1)*, pp. 1-3, 2009
9. Macvittie L. A., "XAML in a Nutshell: A desktop Quick Reference (In a Nutshell O'Reilly))", O'Reilly Media Inc, USA, 2006

Discover the Wonderful World of Plants with the Help of Smart Devices

**Cosmin Irimia, Mihai Costandache, Mădălin Matei, Matei Lipan,
Ştefan Romanescu, Adrian Iftene**

Faculty of Computer Science, “Alexandru Ioan-Cuza” University of Iaşi, Romania

{cosmin.irimia, mihai.costandache, andrei.matei, radu.lipan,
stefan.romanescu, adiftene}@info.uaic.ro

DOI: 10.37789/rochi.2020.1.1.12

ABSTRACT

“Augmented Reality in Botanical Garden” (ARGB) is the name of the application we developed, which brings visitors closer to the Botanical Garden “Anastasie Fătu” from Iaşi. Even if they want to search for a plant to know where they can find it or maybe they are interested in knowing the garden and its sections, visitors can just access the app and get all the help they need. More than this, the app provides enhanced features for the people responsible for the maintenance of the garden such as the possibility to generate placards for different plants during different plant exhibitions.

Author Keywords

Android; iOS; Mobile app; Web app; Maps; Orientation and movement outside buildings.

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces. H.3.2. Information Storage and Retrieval: Information Storage.

General Terms

Human Factors; Design.

INTRODUCTION

Many people believe that computers improve our daily lives, but in many ways, computers, phones, and gadgets that surround us do nothing but isolate us further.

Probably the best example to support this is Virtual Reality (VR) where the big “benefit” is that the real world is replaced by an imaginary, virtual, computer-generated one where you can do almost what you want and be who you want. Instead, Augmented Reality (AR) uses computers to increase the richness of the real world [1], [4-6], [9-14]. It differs from virtual reality in that it does not attempt to replace the real one but only to add various improvements to it.

Our prototype guide through the Iasi Botanical Garden overlaps various information over the real world depending on the place and the images captured by the user’s phone. This method will provide additional information than the tables already around the ecosystem built in the Botanical Garden. Besides information about each plant found in the garden, it also allows the identification of bird sounds and can even guide users through predetermined routes, designed to provide a better experience.

EXISTING SOLUTIONS

PlantSnap - Plant Identifier

PlantSnap[1] is an iOS application that allows only through a simple photo to identify up to 625,000 plants and trees and provides information about them in over 30 languages (see Figure 1).

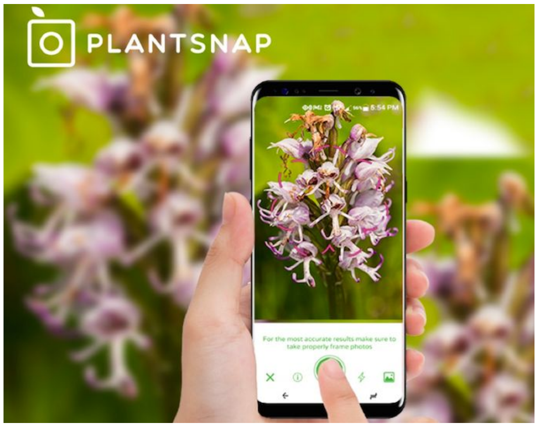


Figure 1. PlantSnap Application

Compared to our application, it has a much larger database

[1] https://play.google.com/store/apps/details?id=com.fws.plantsnap2

because the number of plants growing in the Iasi Botanical Garden is much smaller, but it does not offer any part of the AR for garden paths nor the identification of bird sounds.

Garden Compass Plant/Disease Identifier[2]

This is one of the best plant identification apps, as it does double duty: it can identify both the plant itself and assist you in identifying diseases that may be plaguing it (see Figure 2). As noted in some of our other problem-related articles, it is very important to identify diseases properly in order to deal with them appropriately, without causing more harm to the plant.

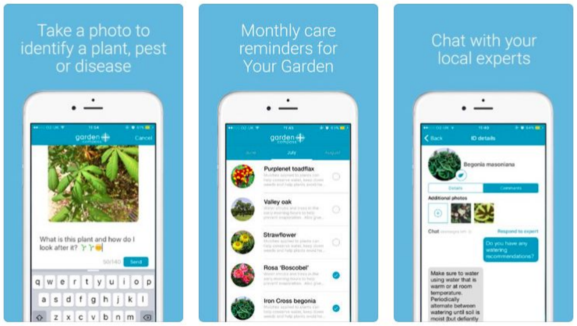

Take a photo to identify a plant, pest or disease
Monthly care reminders for Your Garden
Chat with your local experts

Figure 2. Garden Compass Plant Application

This app is useful, both for amateurs and experts, especially when it comes to combating diseases, and has an interesting technology built in but will not rise at the level of expectation we have with our needs.

PictureThis - Plant Identifier

PictureThis[3] is also an iOS application.

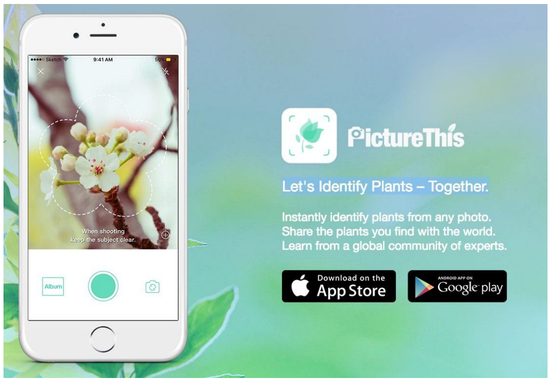

PictureThis
Let's Identify Plants – Together.
Instantly identify plants from any photo.
Share the plants you find with the world.
Learn from a global community of experts.
Download on the App Store
Google play

Figure 3. PictureThis Application

This allows with a simple picture to identify plants and trees, similar to the Plant Snap app, but has a much smaller database (see Figure 3). This can provide information, tell the user how to care for that plant and more than that, it allows posting pictures with plants in the community so anyone can use them as a wallpaper.

Apart from the fact that it does not offer a sound identification part, this application is prepaid, a big disadvantage compared to the one we built.

BirdNET: Bird sound identification

BirdNET[4] is an Android application that allows the user to continuously record the sounds of the environment, and select a piece of sound to predict which species of bird to sing (see Figure 4).

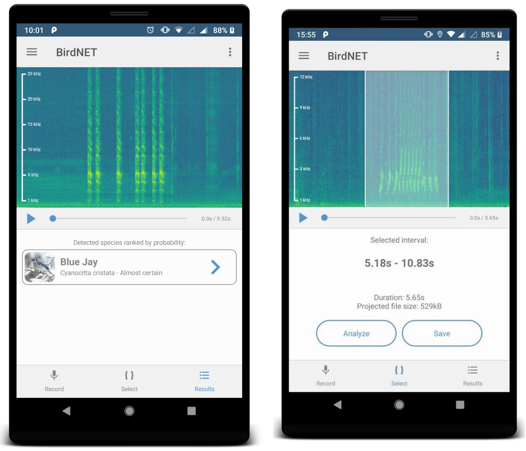

BirdNET
Detected species ranked by probability:
Blue Jay
Cyanocitta cristata - Almost certain
Selected interval:
5.18s - 10.83s
Duration: 5.65s
Projected file size: 529kB
Analyze
Save
Record
Select
Results

Figure 4. BirdNET Application

The application is free but it lacks many things. For example, it can not automatically identify the sound of the birds in the recording, the user has to select only a small piece, and also records the sounds in the environment continuously, which is a privacy issue.

EXISTING WORK

In [8], the authors analyzed the functions of botanical gardens, classified 30 different botanical gardens from all around the world according to purposes ordered by priority and studied the recreational functions of the Nezahat Gökyiğit Botanical Garden in Turkey. Several surveys on the topic of botanical gardens visitors' preferences are referenced by the authors and they also conducted their own. The authors were interested in aspects such as the seasons preferred by the visitors, the frequency of the visits or the reasons people enjoy being in a botanical garden.

In [3], the authors noticed that botanical gardens often focus on biological purposes, e.g., the need to preserve biodiversity, giving less importance to social and

[2] https://www.farms.com/agriculture-apps/gardening/garden-compass-plant-disease-identifier

[3] https://apps.apple.com/us/app/picturethis-plant-identifier/id1252497129

[4] https://play.google.com/store/apps/details?id=de.tu_ chemnitz.mi.kahst.birdnet&hl=en_US

educational aims. Educationally speaking, emotions and cognition together bring great benefits to the learning experience. There was proposed the concept of "useful botanical garden", based on three dimensions: sensibility (features that lead to welcoming comfort, safety and homeostasis), functionality (features that encourage interactions and cognitive processes) and rationality (visitors' reflection on values such as care and authority and how the values relate to the elements present in the botanical garden and the visitors' emotions). The presented ideas have implications for the design of botanical garden spaces and for the formative process based on garden spaces, visitor expectations and interactive activities.

In [7], the author approached the role of Mediterranean botanical gardens in the plant life of the region. They facilitate studies, introduction of new material, conservation and recovery and reintroduction of threatened species. The author noticed an imbalanced distribution of botanical gardens; more specifically, in the southern and eastern Mediterranean countries the botanical gardens have a less important role. Other problems were also analyzed, such as the fact that some botanical gardens cannot accommodate conservation collections of growing plants because they are too old and/or small. Several solutions to the identified problems were proposed.

In [2], the author described how the botanical gardens contribute to university education. The botanical gardens present diverse collections of plants and are accessible for frequent visits. Some skills in botany and/or related fields of activity are best taught with living plants, which make a botanical garden a great choice for learning (even if several activities that are performed in natural areas are not available here). Botanical gardens are also relaxing spaces that encourage learning and creativity. The author noted that the botanical gardens are underutilized, due to several reasons, e.g., proximity issues, lacking or inaccurate interpretation and labels.

FUNCTIONAL REQUIREMENTS

The functional requirements were established together with the experts from Botanical Garden of Iaşi, as they know best what problems they encounter in their work and also they can anticipate pretty accurately what a visitor wants. A main feature of the proposed solution from this paper is the capability of the system to store and collect different data for a series of plants. This part will be done with the help of an web app component:

- An employee from botanical garden will be delegated to insert data about the garden's plants;
- The employee will log in using an administrative account in the web application;
- In the application the user can add a new plant using the following properties: *Plant's name, species, sub-species in latin, a description, qr code that will be near plant plate and an expiration time to make this plant inactive*;
- After all the mandatory fields have been completed the user can save the record;
- The user can use the application to display a list with all of the records, to edit and delete them.

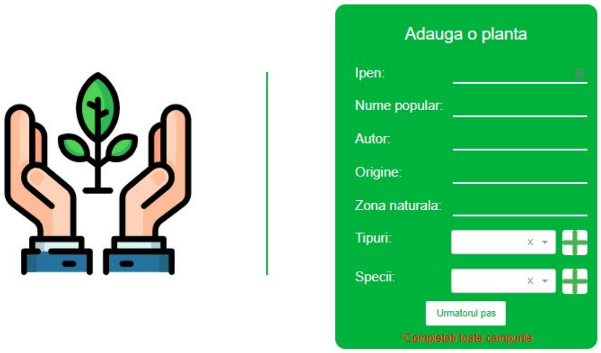


Figure 5. Web Application Component

Another component that will be delivered for visitors with the help of the garden's employee will be the one that guides them through the garden. This will be done with a mobile application component:

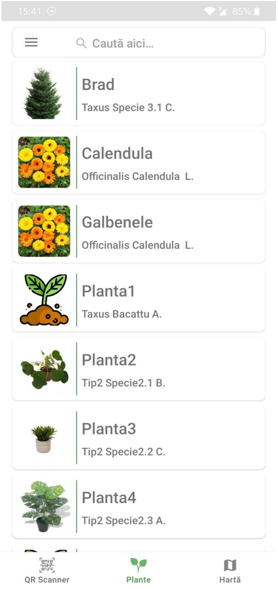
- An employee opens the app in admin mode;
- He/She opens "AR path generator" feature;
- The app opens the camera and detects the writing;
- With this map the app draws steps towards a tree or a plant;
- The admin saves the path and makes it accessible for the visitors.

The interaction between the garden's environment and the visitors will be made through a mobile application. The application is supported both on Android and iOS devices. A capability of this application will be to identify plants or trees with or without an QR code attached (see Figure 6). This will be done as follow:

- The user should install, firstly, the application from the dedicated marked (Google play or App Store);
- The user opens the application, selects from the menu "QR detector" and points the phone's camera to a QR code or to plant;
- After the image is processed, the user can see the description entered by the garden's employee or from the service that identifies images.

Apart from flowers, plants and trees the user can interact with the birds from the botanical garden. This is done as:

- The user opens the application;
- Selects from the menu "Bird sound detection" feature and starts a sound recording session;
- Sends the recording to the Bird Recognition API;
- The API responds with the identified species (and perhaps other useful information);
- The user sends feedback regarding the received information, if he/she wants.



15:41 85%

Caută aici...

Brad
Taxus Specie 3.1 C.

Calendula
Officinalis Calendula L.

Galbenele
Officinalis Calendula L.

Planta1
Taxus Bacattu A.

Planta2
Tip2 Specie2.1 B.

Planta3
Tip2 Specie2.2 C.

Planta4
Tip2 Specie2.3 A.

Figure 6. Mobile Application

The feedback feature is a possible feature, we will think about adding it. The main idea is that if the user is a specialist/bird enthusiast, he/she might know if the API is correct or not regarding a certain audio recording. The feedback could be used by the API. For example, an administrator checks the received feedback and if the user is correct, his/her response is used to improve the trained model. Another possible use would be to compute various metrics. Again, it would be recommended to have the users' responses checked by an admin.

NON-FUNCTIONAL REQUIREMENTS

Besides the functional requirements this application should fulfill non-functional requirements in order to be used efficiently. Thinking of these requirements usually implies technical knowledge, but the experts from the Botanical Garden of Iaşi also helped, as they are aware of the basic requirements of such a complex app too.

Performance

The application should respond fast enough to the image and sound recognition taking in consideration that the number of visitors in botanical gardens is growing every year. This should take into account the number of flower/trees species.

Security

Security measures should be taken into consideration in the admin mode when an employee enters data, so an external attacker will not alterate anything.

Scalability

Given the fact that a feature like bird sound recognition can be used much more frequently than others this should be easy to scale.

Recoverability

Because the path recording is a time consuming process for garden's employees there should be a way to recover these in case of a failure (e.g regular back ups).

PROPOSED SOLUTION

The ARGB Application has 4 main modules. The system architecture is presented in Figure 7.

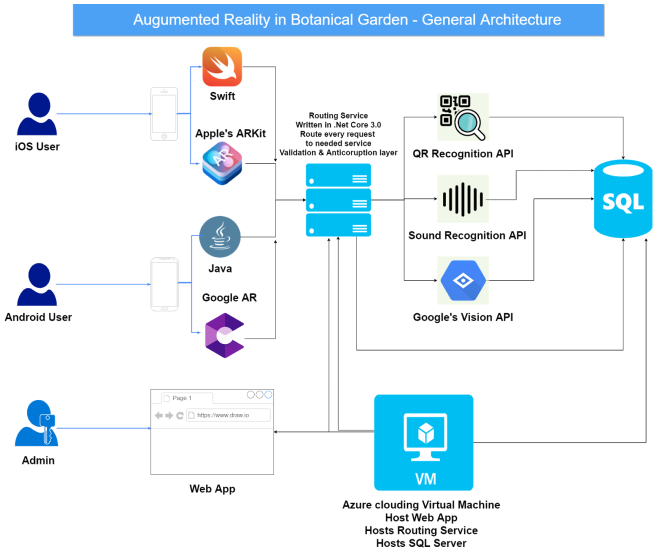


Figure 7. "ARGB" System Architecture

Each of these modules is presented in detail in the following paragraphs.

Android App

This module is the Android part of our application, it is written in Java and builded for phones that run Android 5[5] (Lollipop) or higher (see Figure 8). When the app will be opened the user will get a prompt asking for Camera Permission. If users deny, the app will close itself, else the app will start and will search for an AR Core App installed on the phone. If the user does not have it already installed,

[5] https://www.android.com/versions/lollipop-5-0/

it will get redirected to Google Play Store to install it. If the AR Core is already present, the app will start just fine. In the diagram below, we show the states of the app during the start operation. After the app opens, the user has a camera view right in front of his eyes and with this, he/she can explore the surroundings.

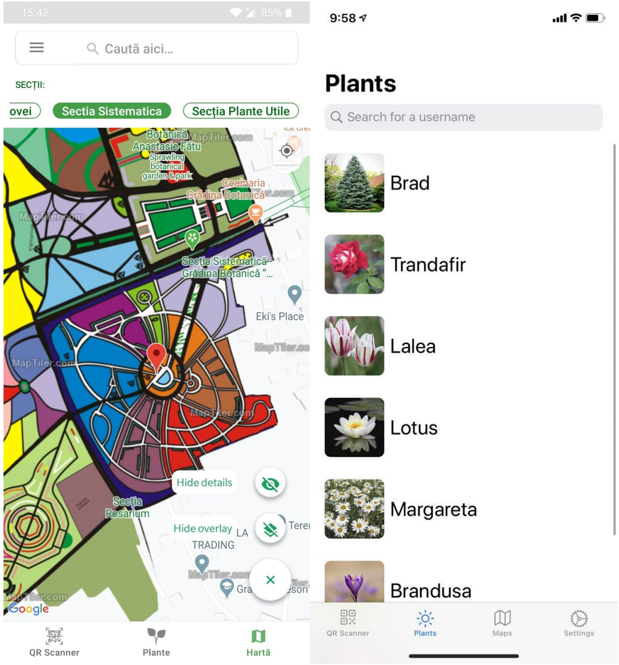


Figure 8. Front page of the "ARGBA" Android App

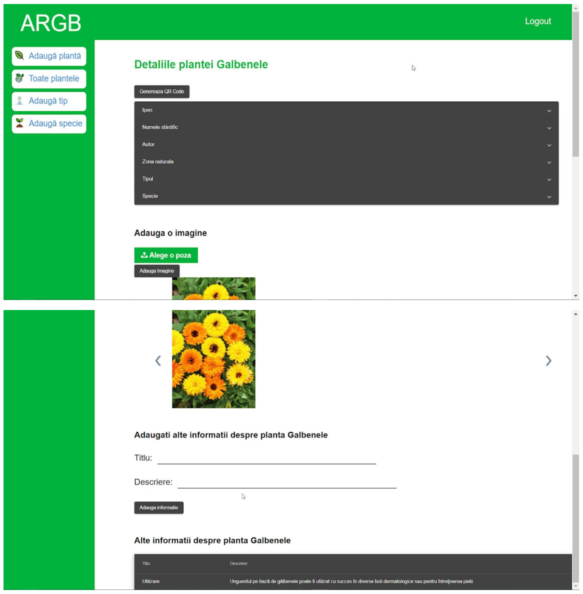
The user has the option to scan an QR Code to get an AR Model of a Iaşi Botanical Garden supported plant and all the information about it or it can scan any plant to get all the information about it but not the AR Model. Also, the app can give the user directions to the Iaşi Botanical Garden for an interesting virtual tour experience. On the other side, the user can record sounds from surroundings, can trim them and identify any bird supported by the Iaşi Botanical Garden application.

iOS App

This module came as a necessity because more and more users have a smartphone that runs iOS. This should run as the Android version as much as the platform gives as this capacity. The application will use Apple's ARKit[6] with the newest SpriteKit[7] framework and will need user's consent in order to use his Camera or the GPS sensor. Like the Android Application, the iOS one works in a similar way, it has a list of plants, a map with all the sections of the Botanical Garden and a QR Scanner that is responsible for recognition of the plates that are all over the garden. The user has also the possibility to change some of the preference settings. Integrated deep into the both applications is the sound recognition module. The user has the possibility to start an audio recording of the surroundings and after he/she has captured the sound of the bird he/she wants to recognize he/she has the ability to crop it and send it to the servers for further processing that will determine which bird he listened to.

Web App

This module is the Admin part of our application, it is a web app written in Angular[8] (see Figure 9). This part of the application will be available only for the admin who will create an account based on which they will be authenticated.



Titlu	Descriere
Utilizare	Unguentul pe bază de gălbenele poate fi utilizat cu succes în diverse boli dermatologice sau pentru întreţinerea pielii.

Figure 9. Front page of the Web App "ARGB"

The admin will have the option to add, edit and delete different types of plants and birds that can be found in the Iaşi Botanical Garden. This module has also a submodule to authorize every non-GET call and make the application inaccessible to anyone who is not an admin. The main page of the application has a simple and clean interface that makes interaction as simple and pleasant as possible.

Sound Recognition

This module is mainly made of an API. The API receives sound recordings and responds with useful information (essentially, the identified species). The app will present the

[6] https://developer.apple.com/augmented-reality/
[7] https://developer.apple.com/spritekit/
[8] https://angular.io/

received information in an attractive and clear manner. We will consider offering the user the possibility to send feedback to the API. The feedback could be used to improve the trained model or to realize several statistics. Feedback is a nice-to-have feature, as we progress with the app development, we will consider if it is worth introducing it and how it should be used (there are several drawbacks - an admin should check the received feedback, it might complicate the interaction between the app and the API, etc.). The module is written in Python and requires several installed modules, such as LibROSA, TensorFlow (with GPU support), Scikit-learn and Flask. It is worth mentioning that the module is developed on Windows and requires a tool, FFmpeg, that enables LibROSA to process more audio formats. There may be similar tools in this context for Windows and other platforms. The mentioned Python modules and their roles in the module are described next.

LibROSA[9] provides several features in audio analysis. In the context of our module, it extracts features from the audio files and saves them in a format that allows us to do the machine learning work. More exactly, we create mel-spectrograms, with parameter values that are chosen based on the default values in LibROSA or experimentally/empirically. LibROSA allows us to model the spectrograms in great detail. For "reading"/"writing" the mel-spectrograms "from/to" files we used Matplotlib[10] and NumPy[11].

TensorFlow[12] (mostly the included Keras API[13]) is used for machine learning tasks. The models are based on neural networks. At installation, we chose an option with support for GPU, in order to get greater speed at various stages in the machine learning processes. The speed is important especially when it comes to the training stage. A trained model is saved and deployed, i.e., it is ready to classify audio recordings received through our API.

Scikit-learn[14] is used for preprocessing. However, it could

have been used for the actual machine learning processes (but it is more appropriate for other datasets/tasks), as it provides decision trees, support vector machines, etc.

Flask[15] builds the API itself, as it allows us to specify routes and perform request/response tasks. We were able to decide what the user will send (mainly, sound recordings but additional information may be provided) and what will be the response from the API (the identified species but also relevant information, taken from a taxonomy).

The module is built in such a way that it allows extensions, so in the future it may be used to classify not just birds. However, the extensions require new data, new machine learning processes to be run (essentially, model training) and small configuration changes in the API.

The module is currently "work in progress", as getting a good training dataset is a challenging task. So far, the work (the API and the ML-related processes) has been done on several sample files, provided by the courtesy of Botanical Garden of Iaşi. We intend to get more files, ideally from the Botanical Garden, in order to create a proper dataset and perform the actual training, for a model that can be deployed.

USABILITY TESTING

In order to be able to analyze whether the objectives listed at the end of the first chapter were achieved, we resorted to performing some usability tests. Usability tests consist of checking a website or a mobile application by potential real users. They analyze elements such as the content, navigation and structure of the application and will note its strengths and weaknesses. In this sense, we have prepared three usage scenarios (described below) that we have offered to several users. After going through these scenarios, we collected their opinions on the application as a whole using a form.

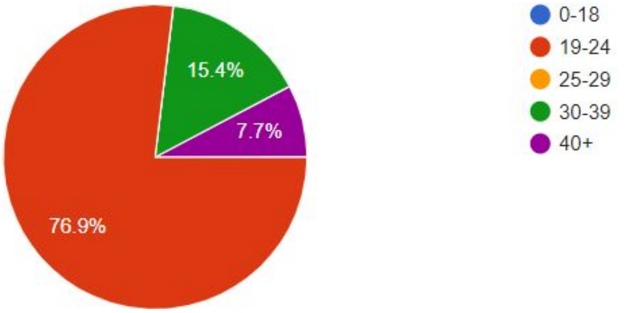


Figure 10. Age distribution of participants in usability tests

In Figure 10 we can see the age distribution of these participants. After completing the three scenarios presented, users were asked to express their opinion on 12 aspects of the application. In the end, we asked them to give an overall rating to the application in its current state, but also to specify how interested they would be in such an application if it had all the functionalities they wanted.

Overall, users were very satisfied with the current way of interacting with the application, the average of the scores

[9] https://github.com/librosa/librosa

[10] https://matplotlib.org/

[11] https://numpy.org/

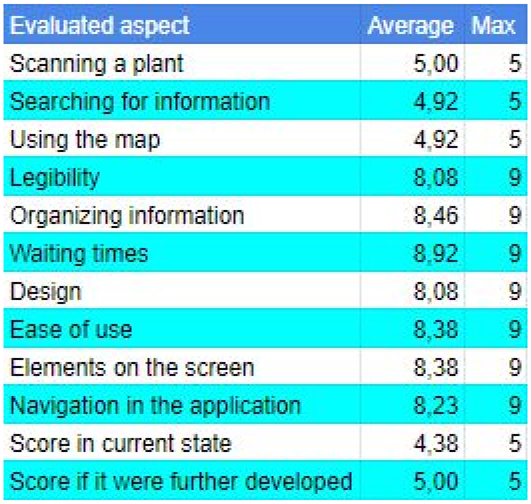
[12] https://www.tensorflow.org/

[13] https://keras.io/

[14] https://scikit-learn.org/stable/

[15] https://flask.palletsprojects.com/en/1.1.x/

given by them being over 85% of the maximum possible for each aspect evaluated separately (see Figure 11). The behavior of the application in each of the scenarios that will be frequently encountered by real users is almost optimal, as can be seen from the first three lines of the other table.



Evaluated aspect	Average	Max
Scanning a plant	5,00	5
Searching for information	4,92	5
Using the map	4,92	5
Legibility	8,08	9
Organizing information	8,46	9
Waiting times	8,92	9
Design	8,08	9
Ease of use	8,38	9
Elements on the screen	8,38	9
Navigation in the application	8,23	9
Score in current state	4,38	5
Score if it were further developed	5,00	5

Figure 11. Average scores for evaluated aspects

Data readability, design and navigation within the application are the main elements that should be improved according to the scores obtained. In order to increase readability, we can resort to another type of text alignment (left-right instead of left), highlighting specialized terms by writing them in italics or even providing this information in audio format. The simple and bright design helps a lot to use the application in the outdoor environment, but on devices with small screens, the subtle delimiting elements are no longer easily observable and the elements of the lists appear almost glued. Navigation options are not constant across all application screens. Inserting a “show on map” button in *SectionInfoActivity* and the menu on the left in all activities and not just in *MainActivity* would solve this problem.

The most appreciated aspect of the application, according to the scores from Figure 12, was the reduced waiting time. This is due both to the use of paging where possible and to the implementation of the automatic caching system and the encapsulation in Livedata objects of all the data to be used by View so that its drawing is not blocked by their waiting.

Next, we asked participants to specify which are the most useful and least useful features of the application (see Figure 13). The map was considered the most important by almost all participants. This was followed by the QR scanner and the caching system, both of which are elements that help to access information as quickly and easily as possible.

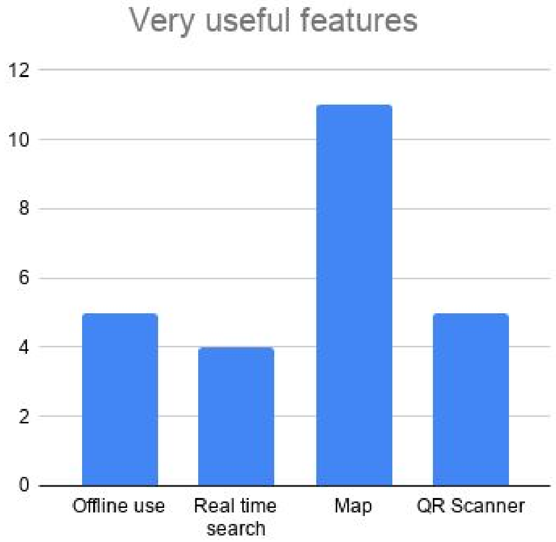


Figure 12. The functionalities considered to be very useful by the participants

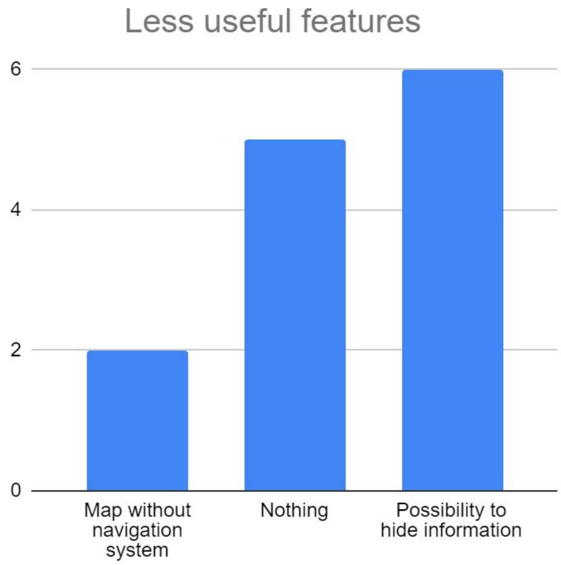


Figure 13. The functionalities considered to be less useful by the participants

In terms of less important functionalities, the possibility to hide information was seen by the participants as being omitted. Their motivation was that, although for a certain plant they consider a type of information to be unimportant and want to hide it, this can lead to the possible loss of interesting data about other plants. At the same time, the map is mentioned here by the participants. This was because these usage tests were done in mid-May, when both the map navigation system using the Google Direction

API and the AR navigation system were not yet implemented. Thus, at the time of the tests, the functionalities offered by the map were quite limited.

The ability to navigate between points of interest on the map or to traverse routes was specified by almost all participants as a desired element.

CONCLUSION

In this paper, we presented how augmented reality can be used in the Botanical Garden, and we show details about application developed by us. The application is complex and has multiple modules.

The Web App is the admin part of the application and it is responsible for adding new plants and birds, managing all the existing information and indexing them all in a simple and elegant way. The Android and the iOS apps are responsible for the end-user interaction. Thus, users can scan plants, can search them, find more information and photos about anything in the garden, can scan a mini-table QR Code to receive more information about something, a map of the garden and all its sections and the bird recognition module embedded inside it. The sound recognition module is a machine learning module that can identify over 20 different species of birds and give the user details about it based on a recorded sound. All this considered, we think our application represents a new way to think and to understand better the existing information from the Botanical Garden of Iași.

For the future, we intend to improve the existing modules and to increase the resources of the sound recognition module.

REFERENCES

1. Bederson, B. B. Audio Augmented Reality: A Prototype Automated Tour Guide. *CHI'95 Mosaic of Creativity, Short papers* (1995), 210-211.
2. Bennett, B. C. Learning in Paradise: The Role of Botanic Gardens in University Education. In *Innovative Strategies for Teaching in the Plant Sciences*, Springer (2014), 213-229
3. Błaszak, M., Rybska, E., Tsivitanidou, O. and Constantinou, C. P. Botanical Gardens for Productive Interplay between Emotions and Cognition. In *MDPI, Concept Paper* (2019), 11(24), 7160; https://doi.org/10.3390/su11247160
4. Cherchi, G., Sorrentino, F. and Scaten, R. AR Turn-by-turn navigation in small urban areas and information browsing. STAG: Smart Tools & Apps for Graphics. *Short papers, Andrea Giachetti (Editor)* (2014), 4 pages.
5. Coates, C. How Museums are using Augmented Reality. *Museum Next. Digital* (2020) https://www.museumnext.com/article/how-museums-are-using-augmented-reality/
6. Ding, M. Augmented Reality in Museums. *Arts Management and Technology Laboratory* (2017), 13 pages.
7. Heywood, V. H. Mediterranean botanic gardens and the introduction and conservation of plant diversity. *Fl. Medit. 25 (Special Issue)* (2015), 103-114, doi: 10.7320/FlMedit25SI.103
8. Karaşah, B. and Var, M. Recreational Functions of Botanical Gardens And Examining Sample of Nezahat Gökyiğit Botanical Garden. In *Proceedings of International Caucasian Forestry Symposium* (2013), 803-809.
9. Iftene, A. and Trandabăţ, D. Enhancing the Attractiveness of Learning through Augmented Reality. *In Proceedings of International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, KES2018, 3-5 September 2018, Belgrade, Serbia. Procedia Computer Science 126 (2018), 166-175.
10. Iftene, A., Trandabăţ, D. and Rădulescu, V. Eye and Voice Control for an Augmented Reality Cooking Experience. *In 24rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*. 16-18 September (2020).
11. Macariu, C., Iftene, A. and Gîfu, D. Learn Chemistry with Augmented Reality. *In 24rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems.* 16-18 September (2020).
12. Păduraru, B. M. and Iftene, A. Tower Defense with Augmented Reality. *Proceedings of the 14th Conference on Human Computer Interaction - RoCHI 2017*, ISSN 2501-9422, ISSN-L 2501-9422, 11-12 September 2017, Craiova, Romania (2017), 113-118.
13. Pinzariu, M. N. and Iftene, A. Sphero - Multiplayer Augmented Game (SMAUG). *In International Conference on Human-Computer Interaction*, 8-9 September 2016, Iasi, Romania (2016), 46-49.
14. Porfireanu, A., Ungurean, A., Dascălu, A., Iftene, A. and Gîfu, D. Smart Museums with Augmented Reality. *In 5th Proceedings of the Conference on Mathematical Foundations of Informatics*. 3-6 July 2019, Iasi, Romania (2019), 285-294.

Secure management and integration system for electrical devices

Bîrsoan Daniel Florin
Technical University of Cluj Napoca
G. Baritiu 26-28
birsoanf@gmail.com

Ștefănuț Teodor
Technical University of Cluj Napoca
G. Baritiu 26-28
teodor.stefanut@cs.utcluj.ro

DOI: 10.37789/rochi.2020.1.1.13

ABSTRACT

The main purpose of an Internet of Things(IoT) network is to make people's work and life easier by providing processes and services as close as possible to their needs. Globally, it is stated that the Internet of Things (IoT) must be available everywhere. As the Internet is almost ubiquitous today, this is not an unreasonable requirement. But to create such a network, it would be necessary for all devices, regardless of the date of creation or manufacturer to be able to be inter-connected to a common platform and made accessible securely through the Internet.

In the current article we are proposing an architecture that responds to this need for the interconnectivity of devices and facilitates secure communication through its components. Through the installation of dedicated board/boards in the desired space and through the connection of the electrical devices on different pins to them, the “objects” are connected to the internet and the user can control their ON/OFF state remotely. So this purpose the proposed architecture features four main components: (1) on-site boards that control the electrical items and the internet connection; (2) a server that orchestrates the communication between all the other components; (3) a web application for electrical items management; (4) a smartwatch application for electrical items control.

Author Keywords

System Security; Modular Software System; Architecture in Software Application, Web Application, Smartwatch Application;

ACM Classification Keywords

•Hardware~Communication hardware, interfaces and storage~Sensors and actuators
•Security and privacy~Software and application security~Web application security
•Social and professional topics~Computing / technology policy~Privacy policies

INTRODUCTION

The Internet of Things (IoT) was first mentioned in 1999 by Kevin Ashton. A common vision in this area is that in the future will be a single global IoT communication network, because the amount of information become huge and there are more and more devices. And another fact is the reliance on the word "things" referring to all the physical objects that surround us.

IoT is an area that have developed a lot in recent years. It is this evolution that has brought more and more IoT solutions develop to the market that has created customized products and aimed at attracting end customers in their ecosystem. Following this diversity, it has not been possible to operate on the initial architecture model for many years and many architectures and ways of working have emerged to manage the very large volume of data and very heavy network traffic.

Also, the interaction and the way we relate to this area must be very well defined and easy to understand by everyone, because, for even greater popularity, all people regardless of their technical knowledge must be able to use the solutions simply and effectively in order to meet their needs. That is why more and more investors in the smart devices development market are migrating to minimal user interactions with the system or systems that manage their space through various innovations such as applications for virtual assistants, Alexa or Google Assistant, in various smart objects, applications for smartwatches or smart TV applications. We must not forget the fact that often the one that facilitates the access to the devices attached by the user in a system are the hubs. As they become more accessible, they have taken over much of the market for the simple needs of users, especially those with built-in assistants, which also help stand-alone applications to communicate with devices.

Following the minimalist interaction described above, I must highlight the contribution of smartwatches in this direction and their rapid evolution in recent years. The clock, after all, over the years, from its appearance to the present day, has been worn by richer people and people representing the middle or lower class. Everyone has become accustomed to his presence, his behavior, and his usefulness in giving us the exact time. It is this habit that has helped the further development of the field of smartwatches. The first smartwatch[1] appeared in 1972, produced by Hamilton Watch and Electro / Data Inc. which was just a digital representation of time in the form of Arabic numerals. Later, it reached an industry that sells

[1] http://www.mobileindustryreview.com/2016/10/33860.htm

over 2.1 million devices annually, most of which are Apple devices.

The smartwatch has a great advantage in managing personal smart devices compared to the regular phone because it is carried on the hand and often replaces its functionality by 80% so users who have such a device are more tempted and satisfied with applications for managing smart objects on it. Its minimalist design, small screen, limited processor and RAM are so far the biggest disadvantage that blocks it from completely replacing the phone.

Security, which is one of the fundamental problems of IoT systems, has been ensured in the proposed solution through the encryption of all communication using tokens. Usability aspects have been addressed through the development of minimalist and easy-to-understand client applications.

RELATED WORK

The book [4] begins with a consideration that specifies that the IoT domain is becoming as abstract as the "Big Data" domain, and how we relate to it must become increasingly personalized. Like development solutions to problems in this area, in this domain we can no longer operate on the "one size suit all" model for years, and this is described in a very objective manner in the chapter [2].

Studies show that sooner or later a single dedicated IoT network will be needed [4] and that all objects will communicate through it. Of course, converting to IPv6 will be a big step forward in this endeavor because the number of public IPs would increase exponentially [1] and every device or hub in space would have one.

And security is one of the most important aspects of the field. The growing number of devices and their holders requires the encryption of sensitive end-user data. Depending on the type of attack, like man-in-the-middle attack or false node message corruption, both the data sending device and the node/server that manages it must be prevented from stopping communication. A list of such attacks can be found in chapter [6], which also describes possible implementations of solutions for each main attack being encryption, object authentication, Datagram Transport Layer Security, or Information Flow Control.

One of the long-term success criteria of a system is the architectural type chosen to develop it. The big developers in the market for object management services in a smart way do not reveal the whole architecture on levels but only large explanatory diagrams or small portions of text that result in how to do things.

An example compared to the system described in this paper is openHAB, which is a company developing custom IoT solutions. In both systems there is the concept of modularity and decoupling of logic data sources. Another existing solution on the market with which the developed system is similar is the application from Samsung, ie SmartThing. From the structure information provided by the developer on the official page of the application we can learn that it relies heavily on the integration of devices in an external server from where a system kernel provides access to applications for customers. As a communication architecture it would be assumed that they use the Client-Server type, similar to the system described in this paper, because they have endpoints through which data is extracted and they must be called by an application or a third party to provide data or perform tasks on the server.

So the competition is given by the diversity of IoT products and applications/systems. Applications such as SmartThing, openHAB, or Google Home, which have gained a lot of ground in recent years due to their scalability and availability, are the main competitors of the developed solution. The competition, after a careful analysis of the market, is based on IoT devices that have either wireless or Bluetooth in their management and integration as opposed to the system designed by us where devices without these two features can be integrated provided they are connected to a power source and operate on the ON / OFF principle.

Solution	Security	Applications dedicated to the system	Electrical consumption of devices
SmartThing	Yes	Mobile/Web/SmartWatch	No
GoogleHome	Yes	Mobile/Web	Yes
openHab	Yes	Mobile/Desktop	Yes
AFHA	Yes	Web/SmartWatch	Yes

Table 1: Comparison between the current solution and competition on the market

Thus we offer the possibility to automate the spaces without assuming the expenses related to the purchase of new, more expensive devices, which are compatible with a certain system, by integrating our system in the space and connecting the existing objects to it.

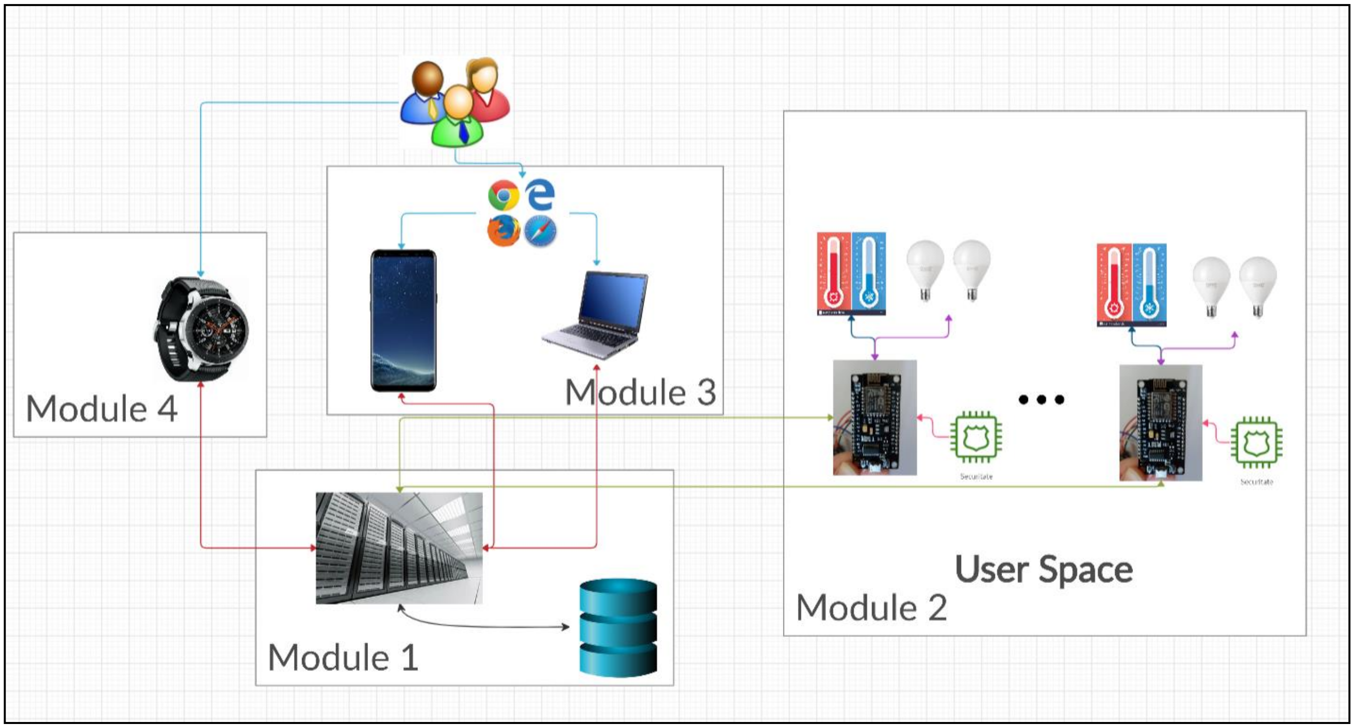


Figure 1: Architecture design.

IMPLEMENTATION

The system consists of four main modules that have been divided according to the functionalities they perform.

The first module, and the center of the system, is the server-side that appears at the bottom of Figure **1**. It is responsible for interconnecting the other components and also responsible for generating the authentication keys for the system. Also in this module is added the database where all the data from the applications are persistently maintained, from the authentication data, where the password and the pin are encrypted with a sha256, to the sensors and devices data from spaces attached to the system.

The second module is the hardware part of the users' buildings. On the diagram, it is on the right side and the plate is symbolically represented by a mini-hub and the bulbs and thermometers represent the rest of the connected devices. The module has the responsibility to connect the electrical devices to the central server through a board that serves as a hub. At the same time, it constantly requests data at regular intervals from the server to find out if it is necessary to execute commands on the devices. The board has an authentication system that request a token from the server before sending the user data and all requests, after this point, are accompanied by the key received after authentication. It also at a certain interval transmits data from the person's spaces to the server to maintain the consumption history and the history of biometric data.

The third module is the web application that has been designed in a way that is compatible with multiple screen sizes. This module is responsible for managing action groups that the user can create for their own spaces, with the ability to activate one or more devices at once. It will also display tables with adjustable consumption, temperature, and humidity depending on the chosen building or the desired time interval. On the architecture diagram, it is represented by the phone and the laptop and through the search engines, the developed application will be accessed. In this case, too, a prior authentication will be made and a key will be received and then used in all requests made to the system. This application is designed to be scalable on both mobile and a desktop screen so that it can be accessed by all potential customers regardless of phone brand or of course whether or not it has a computer or laptop.

The fourth module is responsible for activating the action groups defined in the web application. For the implementation of this module, the development of an application dedicated to smartwatches was chosen because the aim was to control the attached components as easily and quickly as possible. The application also benefits from a notification system through which the end-user is notified when the temperature or humidity exceeds a normal threshold in his home to prevent any incident in the person's premises. Notifications are also received when the application runs behind the others and if more come in a short time they are merged by the system into a larger notification where they are displayed in the order in which they came to the smartwatch. Given the increase in quality in the field of these smart accessories, the implementation of the solution was much more affordable, but still, in terms of hardware specifications, smartwatches lag behind phones and do not provide a complete transition from phones to them soon. This module is represented by the Samsung smartwatch on the left side of the diagram. The communication between each module and the main server is made using the internet networks and the http1.1 format and a key generated following an authentication made to the server is attached to it. Also, to make communication more efficient and to facilitate the openness of the system as a functional requirement, a common language based on the JSON data structure was implemented. Even if one of the components disconnects from the system, it will still be able to operate based on these formats.

The diagram also describes how users interact with the system. It is observed that they have access to the smartwatch application and the web application through browser on the phone or laptop or computer.

Server

The need for a central server is given by the control of the large data flow that the attached components manage, leaving them to focus on managing their main functionalities. It links the persistent part with the user data and the modules to which it provides processed or raw information depending on the need of the place where they are requested. For the easy integration with the rest of the modules and the use, without explicit direct intervention from the programmer, of several threads to perform the tasks, a server designed with Java was used in an application such as Spring Boot.

The server is created in a Layered architectural style with 3 separate levels for the database part, for service logic, and for the connection to the outside through web service.

Security

The security part is provided by the server by generating a token after successful authentication. Once the modules have made this registration and have the token they attach it to the header part of the http1.1 in each request they make to the server. The token part was implemented using the io.jsonwebtoken library from which with the help of an ally generated key for an encryption algorithm a string is obtained through which the personal information about the user who made the request is hidden by the attacker. The algorithm used to generate a token is HMAC which generates a hash function after a private key, for creating the message in the hash sha512 is used for a longer key length and for increased security.

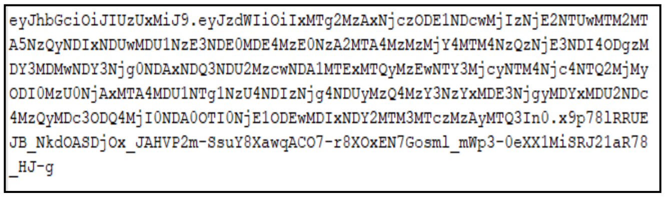
However, knowing the problems that could arise related to the encryption algorithm and the fact that it could be broken and the attacker could take over the secret information entered in the token, we added another level of encryption for the user's data. Before adding the information for encryption, the text is transformed into an integer, the integer is corresponding to the character values in the ASCII table. To secure the text we also used a vector of characters with a randomly chosen size between 20 and 40, the range was chosen arbitrarily to generate a key large enough. Then scroll through the text transformed into ASCII and if the corresponding number in the table is below 9 add two zeros before the number, if the number is less than 100 only add a zero before it and if it is greater than 100 it will not be added any string. After the newly created value, an extension represented by the above fixed-length vector, called the jump, with random values, will be added. The next number in the original string will again generate a vector with the same size but different values and will be added to the new string in the original text. All steps will be repeated until all text is consumed. And finally, we will move on to the classic encryption of the new format string.

To maintain a constant key to generate the token and the size was chosen for the string that is added after the ASCII values, we used the Singleton design pattern in which the class responsible for generating, validating and decrypting the token messages is instantiated once and the data from class is reused.

However, in order not to risk reusing the older authentication keys, the key and the vector size will change every 12 hours with a cron job set on the Spring Boot application. This was only possible by allowing the spring application to be scheduled in the main class with a specific annotation.

And for the decryption part of the message will be considered the first 3 digits of the final text as the first letter with the corresponding value in the ASCII table, after knowing the size of the string of numbers added after it will be iterated over that number of values to the next value of 3 digits which is the second corresponding letter in the ASCII table. Repeat the steps described above until the size of the final message is reached and the original text is obtained. Pre-encryption of the original message brings a security bonus assuming that the attacker could decrypt the hash function used in the token.

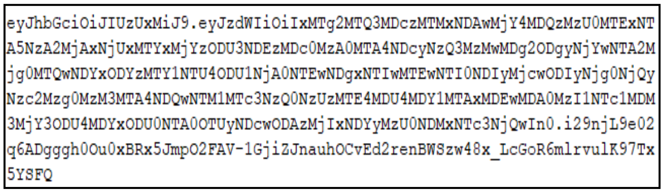
Therefore, on two successive calls to generate a token for use in system requests by a user named "vas.ile1", it will look like this for the first time.



```
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIxMTg2MzAxNjczODE1NDcwMjIzNjE2NTUwMTM2MT
A5NzQyNDIxNDUwMDU1NzE3NDE0MDE4MzE0NzA2MTA4MzMzMjY4MTM4NzQzNjE3NDI4ODgzM
DY3MDMwNDY3Njg0NDAxNDQ3NDU2MzcwNDA1MTExMTQyMzEwNTY3MjcyNTM4Njc4NTQ2MjMy
ODI0MzU0NjAxMTA4MDU1NTg1NzU4NDIzNjg4NDUyMzQ4MzY3NzYxMDE3NjgyMDYxMDU2NDc
4MzQyMDc3ODQ4MjI0NDA0OTI0NjE1ODEwMDIxNDY2MTM3MTczMzAyMTQ3In0.x9p78lRRUE
JB_NkdOASDjOx_JAHVP2m-SsuY8XawqACO7-r8XOxEN7Gosml_mWp3-0eXX1MiSRJ21aR78
_HJ-g
```

Figure 3. The first authentication key for the name vas.ile1.

And for a second request, the authentication key to requests to the server will look like this, different from the first authentication.



```
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiIxMTg2MTQ3MDczMTMxNDAwMjY4MDQzMzU0MTExNT
A5NzA2MjAxNjUxMTYxMjYzODU3NDEzMDc0MzA0MTA4NDcyNzQ3MzMwMDg2ODgyNjYwNTA2M
jg0MTQwNDYxODYzMTY1NTU4ODU1NjA0NTEwNDgxNTIwMTEwNTI0NDIyMjcwODIyNjg0NjQy
Nzc2Mzg0MzM3MTA4NDQwNTM1MTc3NzQ0NzUzMTE4MDU4MDY1MTAxMDEwMDA0MzI1NTc1MDM
3MjY3ODU4MDYxODU0NTA0OTUyNDcwODAzMjIxNDYyMzU0NDMxNTc3NjQwIn0.i29njL9e02
q6ADgggh0Ou0xBRx5JmpO2FAV-1GjiZJnauhOCvEd2renBWSzw48x_LcGoR6mlrvulK97Tx
5YSFQ
```

Figure 4. The second authentication key for the name vas.ile1.

It is also worth mentioning that during the 12 hours until the renewal of the key for generating token encryption functions both generated examples remain valid and the system will treat them equally for the user who owns them.

Hardware integration

A NodeMcu v3 - Lolin development board containing an ESP8266 chip was used to integrate the devices and sensors. The chip facilitates the connection of the board to an internet network thus creating a connection between the main server and it.

A client-server architecture was used to communicate and exchange information between the development board and the server over the Internet, to the detriment of a publish-subscribe architecture. We went on this so as not to make communication very difficult and not to block certain channels as the second one described did. If we went to the second one a port on the server side was continuously busy to detect certain events and thus the system became limited by the number of ports available on the machine where the server application was running.

In the context of the client-server architecture, two modes of communication with the server module were considered, the one in which the server searches for the board to query and transmit information and in which the development board searches for the server to receive information about the tasks to be performed and sends him the latest sensor readings via http1.1 messages. At this point, we went for the second option to disconnect the server from the boards and not need changes on the server-side when adding new spaces and buildings to a person.

The data is sent in JSON format using the ArduinoJson library created by Blanchon B. which provides support for data serialization in chapter [2] and for their deserialization in chapter [3]. Also in this way, after authentication, the authentication key (token) is taken over and stored in a character vector on the board, being used later in the requests made to the server. The size of the vector with which the data is unearthed from JSON has a fixed length of 800 characters because performance reasons the dynamic allocation vector for the library used to extract the data was removed.

For the reading part of the sensors, the data is taken from a DHT22 sensor that provides temperature and humidity in the form of float variables and then with them a JSON is created and the information is sent.

Each device connected to the server has a unique identifier for the board that is attached to it for a specific space. In the implementation part related to the switching on or off of the connected devices, a list is sent from the server with these identifiers in the form of a string for example "012". The list is deserialized and all the devices that are in the list are turned on and the rest remain off. The list is built according to the needs and settings of the client. The limitation of the hardware system here was to 3 devices but the aim is to add a much larger number of devices to the development possibilities.

Client applications

On the client-side, there are two applications which focus on different functionalities. The first one, implemented as a responsive web application, is focused on managing and viewing device statistics on a computer, laptop, or mobile phone. The other, allows the user to control the devices, more precisely their activation, and also to receive notifications in case of exceeding certain normal values for the owned spaces. The Tizen system has been used for the development of this second application, ensuring compatibility with smartwatches that use this system.

The web application is designed to be scalable on both the desktop and the phone. Also, the web application is built in a high usability mode with big buttons and also written big enough not to make it difficult for the user to perform the tasks.

The security for unauthorized access in web application is primarily held by the index class where page routing is based on URLs. When a user does not have an authentication key set in the local storage part of the browser, they are not allowed to access the page and are redirected from the router to the unauthorized page. This is done by a URL analyzer that looks at how the URL is formed and if it recognizes it, in case of "/" it goes to the authentication page, or in case of "/ client" it checks the authentication key from the local storage and if it exists enter the page. If the key exists and is not valid, the user enters the page, but does not see any data because at the time of requests to the server it rejects them.

Also, on the web security side, XSS type attacks were taken into account, which allow the attacker to run a program inside web pages through which he can extract data about the client and about the traffic on that page. This attack is not limited to inserting code by common means such as fields where data is inserted directly but can also be inserted into tables or menus with multiple selection. For this reason, in each field where data can be inserted, they are considered text by the application. On the tables side, the code for them is generated dynamically so that the attacker cannot insert static code in one of the table options. The same principle applies as in the case of tables with multiple selections. On the design side of all the visual components described above, it was ensured that no line of code that could be run or selected was in the control of the browser, thus blocking access to the application code in a direct way.

The smartwatch app is compatible to run on Tizen 4.0 and higher. It uses a readable menu that contains buttons that have the person's buildings on the first level, on the second level after clicking on a building you can see the person's spaces to the building and on the last level are user-defined action groups attached to that space. The user-defined action groups, with the devices you want to start when activated, are red if the scenario is inactive and the devices are off and green if the devices are on and the scenario is active. When the user presses a green scenario, all devices close and the scenario turns red.

For the authentication part, a minimalist design was created with few elements in order not to visually load the user and to make the interaction with the system as easy as possible. The distance between the text where the information to be entered is specified and the space allocated for input is left for a hidden element where the text specific errors are entered when it occurs. The error display mode and the authentication screen are visible in Figure 5.14. Also, a detail of the design part is that all the elements are centered and expand according to the space available on the clock screen. The application has been designed to have the same design experience on multiple watch sizes and sizes. The menu was stacked with centered and dynamically allocated elements when creating the page. If the elements take up more space than available on the device running the application then a right-hand browser will be autogenerated that will allow them to pass through, such an example can be seen in Figure 5.15 in the clock on the left. Also, each menu has a button fixed at the bottom that allows the user to return to an internal action in case it is wrong, but this action does not affect the values that are already saved that have been selected by the user, ie not remove. But a new press of a menu item and the default move to the next level involves overwriting the values of the user's actions.

Also the smartwatch application will receive notifications about the high temperature or high humidity from the server upon request and will display them to the customer on the watch.

For the notifications part, a multi-threaded solution was used. The application runs on a certain thread and before making the first load, from a cycle in which it remains on, it creates a thread on which checks are started in connection with the customer alert situations. The thread, which contains the task of checking notifications, has a timer that is set to 10 seconds, so it queries if new information has appeared about the status of the client's spaces. If the list is not empty, the resulting text is transformed into a list of notifications and sent to a service that sends them to the customer.

CONCLUSION

Pursuing the level of security as one of the requirements of increasing interest in IoT devices, I focused on this by providing token-based communication in all modules and the project as the first version achieved all its objectives.

It can integrate electrical devices that work on the principle of ON / OFF, provides secure communication between its components, and have also been created in a modular style aiming to decouple the components that make it up and a minimum dependence, only in terms of data format. An additional security level has also been implemented for the token generation by encrypting user data with an algorithm that takes into account the ASCII codes of the characters.

Client applications also provide security when communicating data with the server and are highly user-friendly, simple, and easy to use without the need to perform many steps to perform the desired tasks.

REFERENCES

1) Al-Anqoudi Y. S., „Internet of Things," 1 February 2020. [Interactiv]. Available: https://www.researchgate.net/publication/339383844_Internet_of_Things. [Acces 14 March 2020].

2) Blanchon B., „Serialization tutorial," [Interactiv]. Available: https://arduinojson.org/v6/doc/serialization/. [Accessed 23 January 2020].

3) Blanchon B., „Serialization tutorial," [Interactiv]. Available: https://arduinojson.org/v6/doc/serialization/. [Accessed 23 January 2020].

4) Croes E., Software Architectural Styles in the Internet of Things, Nijmegen: RADBOUD UNIVERSITY NIJMEGEN, 2015.

5) Hassan Q. F., Internet of Things A to Z: Technologies and Applications, Al Manşūrah, Egypt: Wiley-IEEE Press, 2018.

6) Jurcut Anca D., Pasika S. Ranaweera, Lina Xu, „Introduction to IoT Security," in IoT Security, Dublin, John Wiley Sons Ltd. 2019, 2019.

7) Nayak P., „Internet of Things Services, Applications, Issues, and Challanges," in IoT, Hyderabad, India, Gokaraju Rangaraju Institute of Engineering & Technology, 2019, pp. 354-366.

8) Santhakumar R., Subramanian B., „IoT Technology, Applications and Challenges: A Contemporary Survey," Wireless Personal Communications, p. 27, 10 April

RASA Conversational Agent in Romanian for Predefined Microworlds

Bianca Nenciu, Dragos Georgian Corlatescu, Mihai Dascalu
University Politehnica of Bucharest
313 Splaiul Independenței, Bucharest, Romania
nenciu.bianca@gmail.com, {dragos.corlatescu, mihai.dascalu}@upb.ro

DOI: 10.37789/rochi.2020.1.1.14

ABSTRACT

Technology is becoming omnipresent in our lives due to its accessibility and ease of use. Conversational agents facilitate interactions in natural language and are frequently employed to perform repetitive tasks in a specific context. We introduce a conversational agent for Romanian built on top of the open-source RASA framework, capable to communicate in predefined microworlds. Two scenarios were considered, namely: a smart home assistant which interprets commands to IoT devices, and an interactive info-point for our university focusing on providing guidance to students. Several enhancements were considered, including an NLP pre-processing pipeline from spaCy and a knowledge graph implemented using Grakn for conceptualizing the information accessible to the agent. Our agent can quickly classify intents and extract entities with high accuracy for a given microworld (F1-score of 97% for the first microworld and 93% for the second). A survey on 10 users showed high satisfaction in terms of the usefulness and the succinctness of the provided information.

Author Keywords

Conversational agent; Natural Language Understanding; Romanian language; Microworlds.

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

I.2.7 Natural Language Processing: Discourse, Language parsing and understanding, Text analysis

General Terms

Natural Language; Text analysis.

INTRODUCTION

Internet of Things (IoT) is a concept that has grown in popularity during the last years due to its utility; numerous day to day objects, such as lightbulbs, can be connected to the Internet and can be switched on and off remotely. Another advantage of IoT is that the interaction with physical objects can be performed using text or voice commands.

Conversational agents are systems that mimic characteristics of human interactions and can have unstructured conversations usually meant to provide information or entertainment to users. Conversational agents are frequently employed in many domains and businesses, such as customer support, sales, marketing, and counseling.

There are multiple frameworks that can be used for implementing conversational agents, but they are mostly available for wide-spread languages, such as English, French, or German. However, there is little to no support for less-spread languages, such as the Romanian language.

This article introduces a conversation agent for Romanian language, capable to communicate in specific contexts (i.e., microworlds). The agent understands intents from the user's input and responds accordingly in Romanian. Note that the methods presented here can be extended to other languages, as well. A microworld can be described as a part of the entire world where the agent lives. It knows the rules governing this small world, can interact with individuals by following those rules, but going outside this context will dramatically decrease the quality of the conversation or even making it inconsistent. Our conversational agent focuses on two different microworlds: 1) a home assistant responsible for controlling IoT appliances available in user's residence; 2) a university info-point to provide student orientation (e.g., guidance on the location of classes or of academic staff).

Two important aspects need to be taken into account when building conversational agents: a) *intent classification* – i.e., the agent should be able to understand what users are saying or what are their interests; and b) *entities detection* – i.e., the agent needs to detect key components from the user's sentence and request missing information. For example, if the user asks "What will be the temperature tomorrow?" the system should be able to understand that the question is about the temperature and also to extract the entities "temperature" and "tomorrow" so that it can respond with the missing information, the actual temperature.

This paper continues with a presentation of the commonly used frameworks for building a chatbot. The following section describes the used corpora, alongside the method employed for building our agent. The paper continues with results in terms of performance and a user survey, followed by conclusions and future leads meant to improve the overall capabilities of the system.

RELATED WORK

Snips [5] is a lightweight dynamic processing pipeline implemented in Rust [20] and Python. Snips can be easily integrated with IoT devices that have limited local resources. Nenciu et al. [16] have extended its pipeline to provide support for the Romanian language.

RASA [2] is a mature open-source framework which contains two main components: RASA Natural Language Understanding (NLU) and RASA Core. The first component is responsible for intent classification and entity detection, whereas the second is the dialogue engine which can be used to implement the conversational agent.

The approach of identifying the intent, as well as discovering corresponding entities, can be performed separately or together. The state of the art model for this task is DIET (Dual Intent and Entity Transformer) [3] implemented in RASA. The model tackles the two problems together and can be trained six times faster than other models, while ensuring accurate results for intent classification and entity recognition. As seen in Figure 1, the DIET model can also receive as input pretrained word vectors from BERT [7], ConveRT or GloVe [17].

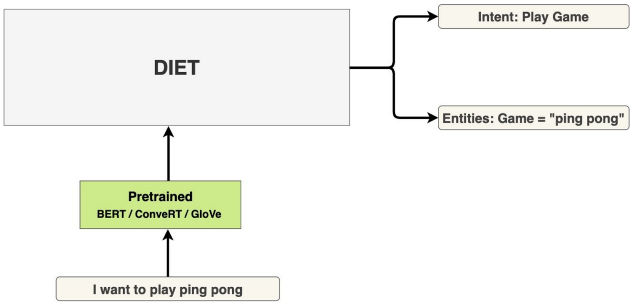


Figure 1. High level illustration of DIET [18].

SpaCy [12] integration is another valuable asset for RASA, especially for low resource languages, such as Romanian. SpaCy is a Natural Language Processing (NLP) tool that provides a common interface for processing all integrated languages. SpaCy can be used to perform part of speech tagging and dependency parsing, whereas these additional insights can help the RASA pipeline provide more accurate results.

In past years, various approaches for simultaneous intent classification and entity recognition have been researched and published. One of the oldest approach was described by Zhang and Wang [26], who used joint models built using Gated Recurrent Unit (GRU) [4], where the hidden state was used for both tasks; their approach managed to outperform the state-of-the-art solutions of that time.

A more recent approach for dual intent classification intent and entity extraction architecture was proposed by Vanzo et al [22] who used both a self-attention mechanism [23] and bidirectional Long Short Term Memory (LSTM) layers [11]. They managed to score better than older versions of Rasa, Dialogflow [8] and LUIS [15].

All previously specified frameworks are focusing on a task-oriented dialogue system. As described by [1] Almansor and Hussain [1], those kinds of systems are solving a specific problem or live in a specific context; this is the reason for being commonly desired by companies. As stated by Sandbank et al. [21], around 80% of interviewed companies want to migrate to this type of solutions in 2020 due to their utility when interacting with a client. The development of such systems is quite straightforward, involving predefined rules and pre-scripted conversations.

For more generic chat bots that are not task-oriented, more advanced solutions are required while relating to their usage scenarios. For example, a chit-chat bot can have issues such as: it can lack specificity, the personality it exposes can be inconsistent, or it can become boring with standard and repetitive responses [25]. State of the art models (e.g., TransferTransfo [24]) for this type of bots consists of approaches using transfer learning and Transformer-based models [23].

METHOD

Corpus

In general, chatbots are task-specific, meaning that they can handle requests from a predefined microworld. This implies that a specific corpus has to be created for each experiment. Two microworlds were explored in this study, namely: a smart home assistant and an interactive info-point for our university. The first corpus was manually created, and it contains 250 sentences on 35 possible intents (see Figure 2 for sample statements). We can further categorize the intents into 12 actions, such as asking about the calendar of the day or controlling home devices. One issue with this corpus was that two intents could have very similar forms, where only one word is different (e.g., "turn the music up" versus "turn the music down"), making it difficult for a model to differentiate between intents.

```
# setTemperature intent
-  Setează temperatura la [roomTemperature](19
   degrees) în [room](bedroom)
-  Poţi creşte temperatura la [roomTemperature](22
   degrees)?
# getRecipe intent
-  Spune-mi reţeta pentru [recipe](pizza).
-  Găseşte-mi reţeta pentru [recipe](clătite).
```

Figure 2. Sample phrases for the first microworld.

The second corpus (see Figure 3) was designed for the university info-point and it consists of both manual and automatically generated sentences. First, we developed a list of entities that can appear in a sentence, such as: name of course subjects (e.g., "Object Oriented Programming", "Electronics"), name of classrooms (e.g., "EG105"), name of teachers, among others. Second, we manually created sentences that had placeholders for the previously mentioned entities. Third, we generated sentences by randomly selecting entities from the specific sets. Given this approach, we generated 80 sentences representing 11 actions which were more different from each other in comparison to the home assistant corpus.

```
## find_schedule_with_course
 -   Unde se desfășoară cursul de [Metode
     Numerice](course)?
 -   Spune-mi, te rog, în ce sală pot participa la
     [Algoritmi Paraleli și Distribuiți](course)
 ## find_schedule_with_class_and_class_type
 -   Unde se ține [laboratorul](class_type) pentru
     grupa [311CB](group_name)?
 -   Unde se ține [cursul](class_type) pentru seria
     [CB](group_name)?
 ## find_schedule_with_course_and_class_and_class_type
 -   Unde se ține [cursul](class_type) de
     [Engleză](course) pentru grupa
     [321CC](group_name)?
```

Figure 3. Sample phrases for the second microworld.

Architecture

The proposed pipeline uses Rasa NLU and corresponding components, and combines them into a new pipeline which offers support for Romanian. We rely on the spaCy model integrated in the ReaderBench framework [6] to perform dependency parsing and part of speech tagging. Figure 4 introduces the overarching pipeline from RASA that relies on spaCy to parse the user query.

The most important components from the NLU engine are the tokenizers – which split the input phrase into smaller semantical units (i.e. words), featurizers – which convert the words into float vectors, and intent classifiers and entity extractors – which in this case are handled all at once by the DIET component. Initial releases of RASA used only a simple CRF (Conditional Random Field) [14], which had problems when the number of training sentences was large (hundreds). DIET has a Transformer-base architecture [23] that uses multiple consecutive CRFs; thus, the new model is no longer susceptible to the initial problems.

Additional relevant Romanian resources integrated in our agent include the DexOnline.ro database, a popular Romanian dictionary. The dictionary itself provides a comprehensive list of word definitions, alongside with word types, popularity, and inflections. Our Romanian resource files consists the following:

- Top 10,000 most used words, together with their corresponding inflections;
- Top 2,000 verbs and lexemes;
- Stop words (i.e., words having no contextual information);
- Randomly generated word lists (i.e., noise used for data augmentation and training the intent classifier);
- Over 1000 of Romanian texts relevant for our microworld scenarios: books, news article, Wikipedia pages.

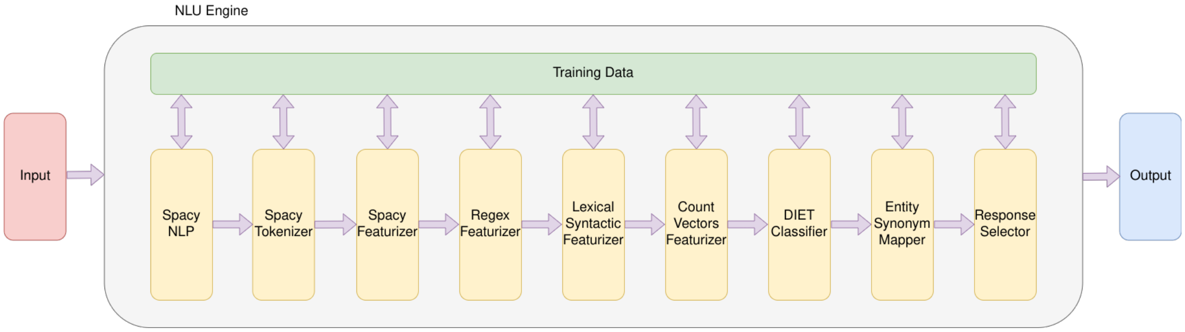


Figure 4. The RASA pipeline integrated with spaCy.

Dialog Management

A dialog manager is responsible for the flow of the dialog between the user and the conversational agent. Figure 5 introduces the steps for the dialog manager, which takes the output of the NLU component, updates the current state of the dialog, the user's history, as well as other important information, and outputs instructions for the response selector.

The input to the dialog manager is a human utterance, converted to its semantic representation by going through the intent classification and the entity extraction process. For example, a question like "Unde găsesc cursul de programare orientată pe obiecte?" (eng. "Where do I find the Object Oriented Programming class?") will be transformed to a query like "find(class='OOP')". As the input is too ambiguous, the dialog management will try to find relevant user information, such as their class name. Furthermore, the knowledge base is queried for information about that specific class and its name. Finally, the agent will output an instruction like "class_location (class_name='2CB', class='OOP', room='PR001')" which is outputted into natural language: "Cursul de programare orientată pe obiecte pentru seria 2CB se ține în sala PR001 la ora 18:00." (eng. "The Object Oriented Programming course for the 2CB series takes place in room PR001 at 18:00.").

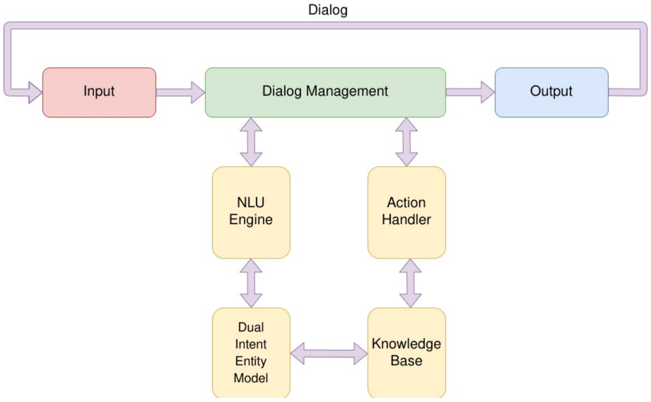


Figure 5. Dialog management architecture.

Intent Classification and Entity Extraction

The first step of every conversation consists of waiting for user's input, which is a characteristic specific to any dialog management system, followed by classifying and extracting entities using DIET. One improvement that can occur at this stage and might be implemented in the future consists of transforming the extracted entities into machine readable representations (e.g., "mâine la 8" / eng. "tomorrow at 8" could be converted to a timestamp).

Context Tracking

User history or previous states must be maintained between queries and replies for conversations to become stateful. For this purpose, a small knowledge base called a "tracker" is built and stored in an in-memory data structure offered by a Redis [19] backend. When a new session starts, a token is randomly generated, which is afterwards passed along with each parsed input. For an even better tracking, the user's username, name, or any form of identification can be passed. In the context of the university chatbot, the tracker can fetch information about courses and other public information after an initial authentication which consists of stating your name.

Response Handling

Response handling is the last, but one of the most important components in building conversational agents. It takes all the information that has been parsed by the NLU engine and previous information held by the tracker, and builds a meaningful answer. There are multiple alternatives in which an agent can produce a reply, which can even involve natural language generation. However, we focused on two simpler techniques: predefined responses and custom actions.

Predefined Responses. The agent can also be trained to associate arrays of predefined responses with intents similar to how it is trained with various input phrases and queries, Moreover, responses do not need to be static, in the sense that the sentences may differ for a set of queries. The simplest strategy for making the conversation more human-like is to define multiple responses for the same intent and randomly select one of those. In addition, placeholders can be automatically replaced based on the extracted entities. For example, if the user greets the agent, then it replies with a greeting as well. A common interaction could be started by the user with a "Hei!" (eng. "Hey!" message, while the agent would respond with "Hei. Cu ce te pot ajuta?" (eng. "Hey! How may I help you?").

Custom Actions. Conversational agents can reply using a custom action implemented in a given programming language that follows an imposed application logic. The usual problem with this approach is that the interactions often seem unnatural, as there is very little nondeterminism or randomness in the output. For this specific reason, custom actions may be combined with predefined responses to reply to the user. Another use case for custom actions is when additional information is needed from the user, or when third party APIs are queried.

Knowledge Representation

Our conversational agent needs to store and retrieve relevant information, as well as the context of a discussion to respond to the user's input. We opted for a non-relational database – Grakn [9] –, an open-source knowledge graph representation that provides an excellent fit for systems operating with highly interconnected data. Grakn provides a concept-level schema which implements the Entity-Relationship model and provides reasoning capabilities. Figure 6 introduces the model corresponding to our second microworld scenario. The agent can perform slot filling tasks by using Graql [10], Grakn's Reasoning and Analytics Query Language, in order to properly continue the conversation with the user.

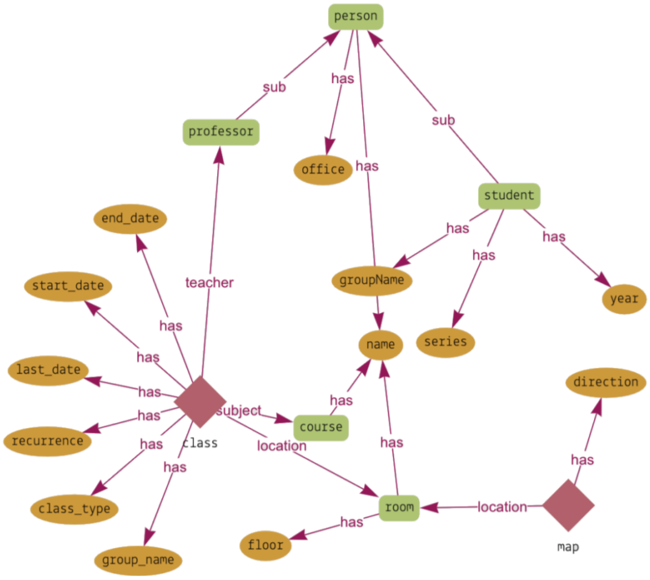


Figure 6. Grakn knowledge graph for the university info-point.

One of the most important and complex parts of the agent while considering the second micro-world relates to navigation queries. The difficulty of answering navigation-related queries comes not only from the absence of a localization system, but also from the fact that a long list of steps may be too difficult to remember. Thus, the agent attempts to describe the destination using the surrounding environment, points of interest, or any other information that the user may already have. Some directions, such as the floor where the room is found, are more important than others, which describe minor details from the surrounding area (see Figure 7).

```
floor sub attribute,
 datatype string;

room sub entity,
 has name,
 has floor,
 plays location;

direction sub attribute,
 datatype string;

map sub relation,
 relates location,
 has direction;

$pr-001 isa  room, has name "PR 001", has floor
"parter";

$map-pr-001 (location: $pr-001) isa map,
 has direction "vis-a-vis de grupurile sanitare",
 has direction "accesibil din holul principal";
```

Figure 7. Schema definition for a point of interest.

Sample conversation

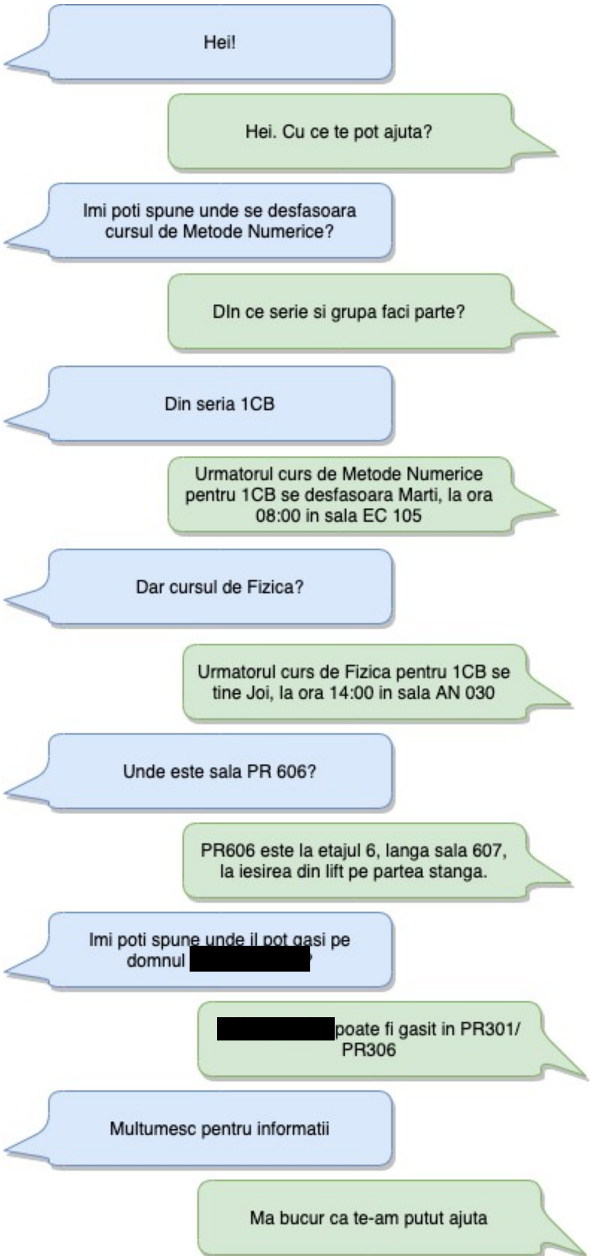
Figure 8 introduces a sample conversation between a user and our conversational agent.

RESULTS

Performance

The training of our agent for each microworld was done on 80% of the corpus; the remaining 20% was used for testing. Although the number of examples is small, this is not a problem due to the characteristics of a microworld which is self-contained, and it considers similar ways to express an intent. The considered metric for assessing the performance of our system was the F1-score.

The first microworld contained 203 phrases belonging to 35 categories, while our test suite contained 94 phrases from the same categories. The system achieved a 97% F1-score with the corresponding confusion matrix from Figure 9. The confusion matrix itself shows promising results and most test phrases were correctly classified. Some pairs of intents which were difficult to classify consist of: “playSong” with “previousSong”, or “volumeUp” with “volumeDown. For example, “previousSong” had the lowest F1 scores of 75%, partially because it is an intent with very few phrases which can be mixed up with “playSong”. This is somewhat expected, given how similar two inputs are, with the only difference between them being “on” versus “off”. In a production environment, we recommend keeping logs for intents and periodically updating the dataset to further increase the system’s accuracy and its performance. This can be either done automatically by using clustering algorithms or by manually curating the logs.



Hei!

Hei. Cu ce te pot ajuta?

Imi poti spune unde se desfasoara cursul de Metode Numerice?

DIn ce serie si grupa faci parte?

Din seria 1CB

Urmatorul curs de Metode Numerice pentru 1CB se desfasoara Marti, la ora 08:00 in sala EC 105

Dar cursul de Fizica?

Urmatorul curs de Fizica pentru 1CB se tine Joi, la ora 14:00 in sala AN 030

Unde este sala PR 606?

PR606 este la etajul 6, langa sala 607, la iesirea din lift pe partea stanga.

Imi poti spune unde il pot gasi pe domnul [redacted]

[redacted] poate fi gasit in PR301/ PR306

Multumesc pentru informatii

Ma bucur ca te-am putut ajuta

Figure 8. Sample conversation between a human (left side) and our agent (right side).

The second microworld contains 80 phrases belonging to 11 intents, and our test set contained 27 phrases covering all intents. This microworld included eight predefined dialogs using 23 responses from 9 categories. The predefined dialogs and responses were used to train the response selector. The agent achieved a 93% F1-score and the corresponding confusion matrix is depicted in Figure 10.

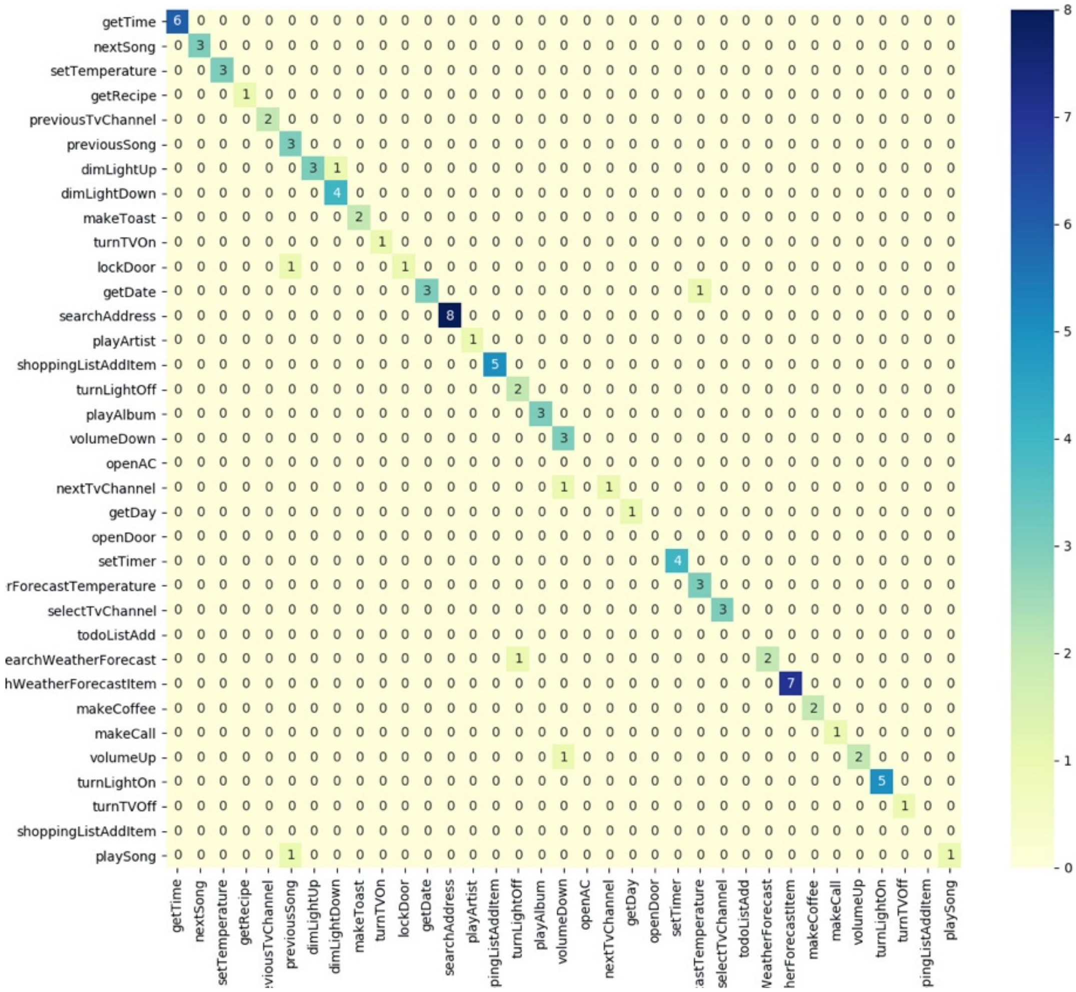


Figure 9. Confusion matrix for the first microworld (home assistant).

Fewer problems are identified since the training dataset was stricter, with fewer categories and with little overlap between the intents. We notice that the "find schedule with course and class and class type" intent is overlapped with "find schedule with class and class type" (without the actual "class"); nevertheless, this does not matter in practice because both intents are handled by the same action. Therefore, the end the user receives the same expected answer.

In addition, the agent has to provide near real-time responses to ensure the flow of the conversation. We achieved response times of less than two milliseconds, which can guarantee the naturalness of the conversation.

User Survey

The NLU engine was evaluated using a small group of 10 undergraduate and Master degree students from our university who queried the conversational agent for information, using the second microworld scenario. The users were afterwards asked to rate their interaction on a Likert scale from 1 to 10 based on the following criteria:

1. How useful was the information? (1 – "Not usefully at all"; 10 – "Extremely useful"; M = 9.00, SD =0.77);
2. How pleasant was the interaction? (1 – "Completely unpleasant"; 10 – "Extremely pleasant interaction"; M = 9.40, SD =0.77);
3. Have you ever considered the other conversation party was a machine? (1 – "I thought I was talking with a person"; 5 – "I cannot say"; 10 representing "I knew that I was talking with a chat bot" M = 5, SD =0.87).

The Intraclass Correlation Coefficient [13] is 0.888, which suggests strong agreement between the replies to the survey. All individuals found the information to be very useful, with a high user satisfaction (9 on a 10-point scale). Most users were satisfied in terms of the quality and the succinctness of the information. The less satisfied users reported they were looking for additional information and they would have preferred to avoid the necessity of a secondary query. While relating to the pleasantness of the conversation, part of the users considered the agent should have included some chit-chat messages, while others considered it to be a very pleasant and the dialog was natural; thus, the low ratings to the last question.

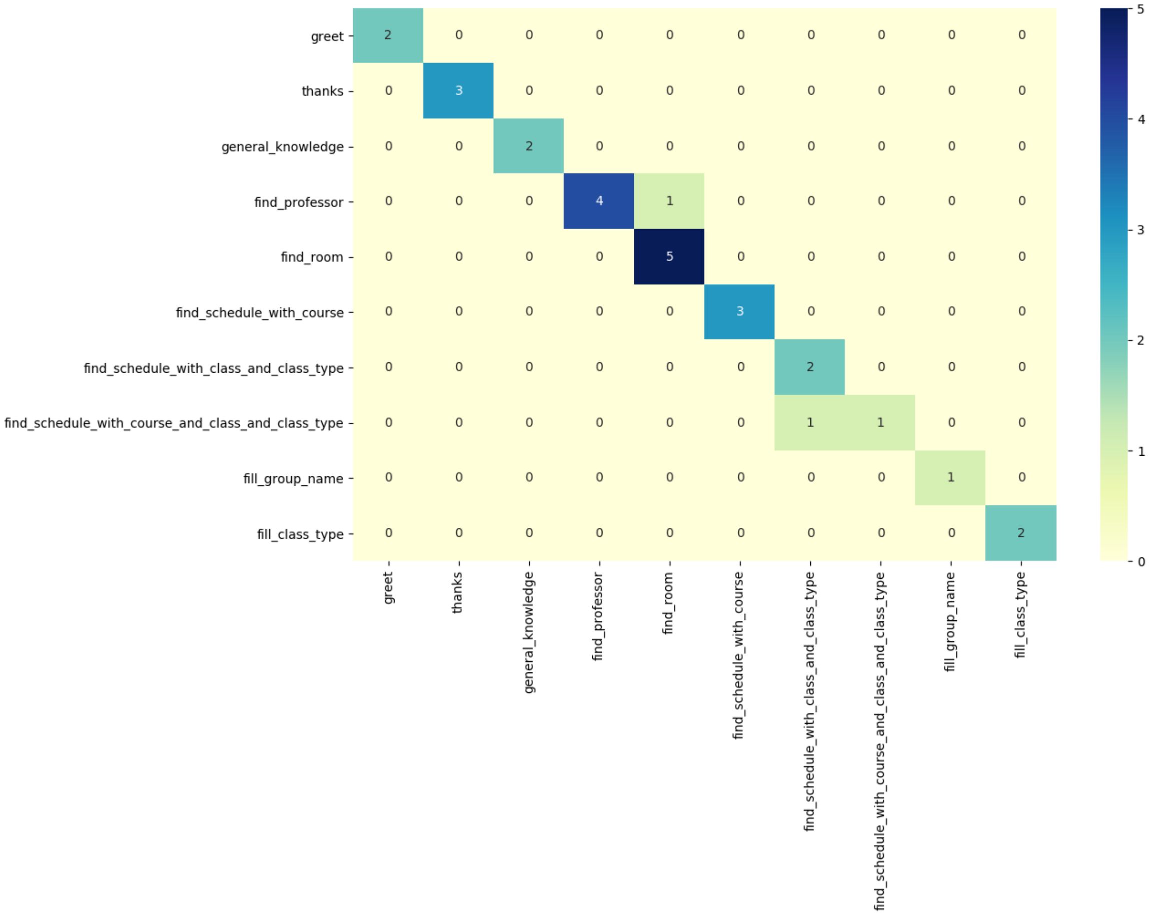


Figure 10. Confusion matrix for the second microworld (university info-point).

CONCLUSION

This paper introduced an NLU engine for Romanian built on top of RASA, capable to quickly classify intents and extract entities with high accuracy for a given microworld. The results are promising for small microworlds that contain a limited number of phrases used to express an intent, and most alternatives are similar. We are working towards building and testing on a larger corpus, which should result in a more general system.

In contrast to close source alternatives, our project runs locally, and it requires few resources after the DIET classifier was trained. Intent classifying with an external service could be easier to implement, but the processing would take a considerably longer time because even the ideal round trip time could already be over 20 times slower than the usual processing time of our engine (1-2ms).

We consider that an important improvement for the NLU engine consists of integrating advanced language models (e.g., a Romanian BERT model), which would expand further the agent's capability across microworlds. In addition, we plan to expand our research to corpus-based architectures to ensure more natural conversations.

Another interesting area of research relates to the classification of multiple intents from a single user statement. In real world situations, we often find ourselves building complex phrases containing multiple actions. The current system returns only the most probable action or none, if no probability is greater than the imposed threshold. In tight correlation to the previous research lead of using the agent in real-life scenarios, our approach was evaluated independently from a speech-to-text engine, which would induce additional errors; however, our training set only contains correct phrases, with no spelling errors. Thus, additional fine-tuning, integration of correction mechanisms, and extensive testing are required.

ACKNOWLEDGMENTS

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 72PCCDI / 2018, ROBIN – "Roboții și Societatea: Sisteme Cognitive pentru Roboți Personali și Vehicule Autonome".

REFERENCES

1. Almansor, E.H. and Hussain, F.K., 2019. Survey on Intelligent Chatbots: State-of-the-Art and Future Research Directions. In Proceedings of the Conference on Complex, Intelligent, and Software Intensive Systems (Sydney, Australia), Springer, 534-–543.
2. Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A., 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.
3. Bunk, T., Varshneya, D., Vlasov, V., and Nichol, A., 2020. DIET: Lightweight Language Understanding for Dialogue Systems. *arXiv preprint arXiv:2004.09936*.
4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
5. Coucke, A., Saade, A., Ball, A., Bluche, T., Caulier, A., Leroy, D., Doumouro, C., Gisselbrecht, T., Caltagirone, F., and Lavril, T., 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.
6. Dascalu, M., Dessus, P., Trausan-Matu, Ş., Bianco, M., and Nardy, A., 2013. ReaderBench, an environment for analyzing text complexity and reading strategies. In Proceedings of the International Conference on Artificial Intelligence in Education (Memphis, TN, United States), Springer, 379–388.
7. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
8. Google, 2020. Dialogflow. Retrieved September 30th 2020 from https://cloud.google.com/dialogflow.
9. Grakn, n.d. Grakn webpage. Retrieved July 27th 2020 from https://grakn.ai/.
10. Graql, n.d. Graql Query Language. Retrieved July 27th 2020 from https://dev.grakn.ai/docs/query/overview.
11. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation 9*, 8, 1735–1780.
12. Honnibal, M. and Montani, I., 2017. spacy 2: Natural language understanding with bloom embeddings. *convolutional neural networks and incremental parsing 7*, 1.
13. Koch, G.G., 1982. Intraclass correlation coefficient. In *Encyclopedia of Statistical Sciences*, S. Kotz and N.L. Johnson Eds. John Wiley & Sons, New York, NY, 213–217.
14. Lafferty, J., McCallum, A., and Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the ICML (Williamstown, MA, USA), Morgan Kaufmann, 282–289.
15. Microsoft, 2020. LUIS. Retrieved September 30th 2020 from https://www.luis.ai/.
16. Nenciu, B., Ruseti, S., and Dascalu, M., 2018. Extracting Actions from Romanian Instructions for IoT Devices. In Proceedings of the 13th Int. Conf. on Linguistic Resources and Tools for Processing Romanian Language (ConsILR 2018) (Iasi, Romania), 168–176.
17. Pennington, J., Socher, R., and Manning, C.D., 2014. Glove: Global vectors for word representation. In Proceedings of the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (Doha, Qatar), Association for Computational Linguistics, 1532–1543.
18. RASA, 2020. Introducing DIET: state-of-the-art architecture that outperforms fine-tuning BERT and is 6X faster to train. Retrieved July 27th 2020 from https://blog.rasa.com/introducing-dual-intent-and-entity-transformer-diet-state-of-the-art-performance-on-a-lightweight-architecture/.
19. Redis, n.d. Redis Homepage. Retrieved July 27th 2020 from https://redis.io.
20. Rust, n.d. Rust documentation. Retrieved July 27th 2020 from https://prev.rust-lang.org/en-US/.
21. Sandbank, T., Shmueli-Scheuer, M., Herzig, J., Konopnicki, D., Richards, J., and Piorkowski, D., 2018. Detecting egregious conversations between customers and virtual agents. In Proceedings the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, Volume 1 (New Orleans, Louisiana), ACL, 1802–1811.
22. Vanzo, A., Bastianelli, E., and Lemon, O., 2019. Hierarchical multi-task natural language understanding for cross-domain conversational ai: HERMIT NLU. *arXiv preprint arXiv:1910.00912*.
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I., 2017. Attention is all you need. In Proceedings of the Advances in neural information processing systems (Long Beach, CA, USA), Curran Associates, Inc., 5998–6008.
24. Wolf, T., Sanh, V., Chaumond, J., and Delangue, C., 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.
25. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J., 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.
26. Zhang, X. and Wang, H., 2016. A joint model of intent determination and slot filling for spoken language understanding. In Proceedings of the IJCAI (New York, New York, USA), AAAI Press / International Joint Conferences on Artificial Intelligence, 2993–2999.

Conversational Agent in Romanian for Storing User Information in a Knowledge Graph

Gabriel Boroghina, Dragos Georgian Corlatescu, Mihai Dascalu
University Politehnica of Bucharest
313 Splaiul Independenței, Bucharest, Romania
gabrielboroghina@outlook.com, {dragos.corlatescu, mihai.dascalu}@upb.ro

DOI: 10.37789/rochi.2020.1.1.15

ABSTRACT

Information surrounds us and keeping track of relevant details can be challenging. Although there are multiple applications to take notes, organize ideas, or set reminders, existing solutions are semantic-agnostic and rely on the user to manually search for desired information by keywords. We propose a novel method to help people store and retrieve such details with ease in Romanian language. Our conversational agent built on top of the RASA framework is capable to extract relevant information from the user's utterances, store them in a persistent knowledge graph, and ultimately, access them when requested. A set of specific intents regarding locations, timestamps, and properties were created and learned by the agent using manually built examples. In addition, an interaction logic based on a knowledge graph was added to enable the storage and retrieval of information, based on the identified semantic components from the input sentences. The performed tests showed a good accuracy for intent detection, and promising results for the sentence parser. Our conversational agent is accessible as a web application which can process text or speech inputs, and responds with a textual representation of the user's memorized facts.

Author Keywords

Natural Language Processing; Conversational agent; Knowledge representation.

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

I.2.7 Natural Language Processing: Discourse, Language parsing and understanding, Text analysis

General Terms

Natural Language; Text analysis.

INTRODUCTION

Conversational agents represent intelligent programs based on Natural Language Processing (NLP) techniques, capable of performing dialogue conversations with a human interlocutor. These agents need to maintain the global state of the conversation and provide responses in relation to this state and the user's messages. Moreover, conversational agents became more and more adopted by the industry, mostly in the form of chatbots that offer guidance to clients with regards to a product or service, but also as virtual assistants capable to perform various tasks at the user request, such as taking notes, setting an appointment, and even controlling smart home devices.

Conversational agents are preferred in a wide range of cases due to their programmability and their scalability for simple and highly repeatable tasks, such as basic technical support. Equally important is the help agents can provide to people who are unfamiliar with the use of a technology, or have deficiencies (such as blindness), by allowing them to simply talk with the agent in order to execute specific tasks. Thus, conversational agents try to push the perceived intelligence of computers and their ability to interact with people further, at an increased pace.

We introduce a memory-assistant conversational agent for Romanian language, capable to remember the user's previously registered information. The agent is capable of processing the user's statements that expose information which should be memorized, and can also react to his/her requests for a specific detail, relying upon the gathered data. The core features consist in detecting the users' intents (whether they wanted to register or to access a specific type of information), and extracting the key facts from their utterances, storing them in a knowledge graph, and exposing them back on demand.

Our agent works with factoid information [15], meaning that it has the ability to detect specific elements inside more complex sentences, and respond to queries such as: "where" (locations of objects or events), "when" (timestamps of events or actions), "which is" (various properties of entities), "which of the" (the instance of an entity that has a certain qualification or function), or "who" (subject of an action or a description). The agent operates in a user-initiative mode, meaning that the user launches the dialogue sessions with a request, waiting for the agent to reply with the corresponding result. Moreover, the information is presented in a concise form to the user, just as it is stored in the knowledge graph; no context planning or natural language generation techniques are used for information retrieval. Users do not have to specify the type of the information to be stored, nor how or where to keep it. The chatbot itself is responsible for interpreting, understanding, and managing data, while the interaction with the user is limited to expressing a fact or a request in natural language (text or speech). In this manner, the agent allows for fast, natural, and intuitive information registration and retrieval.

STATE OF THE ART

Conversational agents are increasingly adopted into different software applications and web platforms. According to Brandtzaeg and Følstad [6], the main benefits of using such agents are their productivity (i.e., obtaining support or necessary information regarding a product or a service), but also entertainment, or social interaction and communication (e.g., chit chat or social chatbots). The bots provide a significantly enhanced user experience and interactivity for performing various tasks than a static user interface. In addition, bots also offer scalability and objectivity in their conversation with the user. Thus, a conversational agent can be perceived as being more agile than a person at providing simple information or helping the user solve standard problems that fit a certain template.

Depending on the complexity of the conversation, two types of agents come into view, according to Jurafsky and Martin [15]: a) simple agents designed to perform tasks at the user's request, called task-oriented dialogue agents, and b) chatbots, which are more complex agents that can maintain longer conversations, which can extend over several replies and can include different interleaved subjects of discussion. In this article, we will focus on the first category.

Conversational agents rely on a dialogue-state architecture [15] that addresses two problems: intent classification and slot filling. Multiple recent approaches try to develop joint models capable to perform both tasks on a single pass through the model. For instance, Liu and Lane [19] proposed an attention-based RNN (Recurrent Neural Network) model which can concurrently perform intent detection and slot filling, for a given token sequence. The model consists of a bidirectional RNN, based on LSTM (Long Short-Term Memory) cells. The model generates slot labels for each token based on a hidden state made up of a forward and a backward state obtained by analyzing the input sequence in both directions. Attention provides additional information about the input sequence through a context vector that is combined with the hidden state before token labelling. Intent detection is performed simultaneously by using the computed hidden states from the bidirectional RNN model. An attention-based encoder-decoder model was also investigated by Liu and Lane [19]; the model integrates all information from the input sequence (encoding), and then generates an output sequence containing the slot labels (decoding).

In contrast to the previous approach, Wang et al. [28] contradict the joint model structure, and propose a bi-model RNN structure, consisting in two connected BiLSTM components, one for intent detection and one for slot filling, that takes advantage of the cross-impact between the two tasks. The two BiLSTM models exchange their hidden states and combine them when generating the output of each task; thus, the intent detection model benefits from the features extracted by the slot filling model and vice versa. A slight variation containing one LSTM decoder on top of each BiLSTM component was also explored, obtaining even better results and surpassing the previous state of the art systems for both tasks.

Dialogue-state architecture can be also used for designing task-oriented dialogue systems [15]. These architectures consist of a more complex pipeline of processing components [15]: automatic speech recognizer, spoken language understanding unit, conversation state tracker, dialogue policy, natural language generation unit, and speech synthesizer. While the first two components act as blackboxes that perform processing tasks over the user's voice input and export texts, the dialogue tracker is a stateful component that maintains at each moment the conversation status, alongside the chat history.

Despite the usefulness and remarkable user experience conversational agents can provide, their creation from scratch can prove to be challenging and time consuming. An open-source framework that comes in handy is RASA [5], which provides Natural Language Understanding (RASA NLU) and Dialogue Management (Rasa Core) features for building intelligent conversational agents with a minimum effort from the programming point of view. The framework is based on a dialogue-state architecture, but supports only the communication through text messages (i.e., it does not integrate speech-to-text and text-to-speech converters). The NLU component has a modular pipeline, instrumental in customizing and tweaking the components involved in processing and interpreting the user's utterances. Tracker objects are used to manage the conversation flow; their role is to maintain a list of events encountered during the communication session, together with relevant facts (slots) exposed by users through their statements.

RASA relies on the DIET (Dual Intent and Entity Transformer) classifier [8] for intent detection. The model obtains a good performance and its training is significantly faster and easier when compared to older models. The architecture can be visualized in Figure 1. The model uses a 2-layer Transformer [27] for condensing the semantic features extracted from each token, and compares the resulting feature vector with the vector corresponding to the gold intent based on a similarity (dot product) metric. A Conditional Random Field (CRF) [17] can be used on top for entity extraction. Pre-trained word vectors from BERT [12] or GloVe [21] can be used to improve the results.

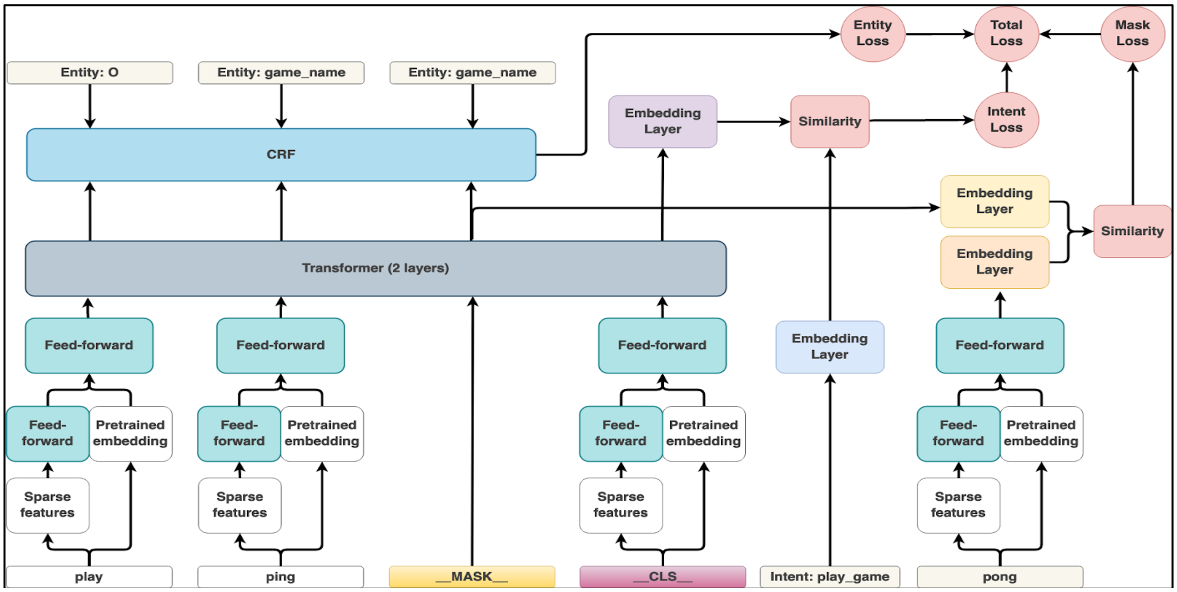


Figure 1. DIET architecture [8].

Data storage is also a central component for our agent. A traditional SQL approach is not suitable because data from the conversation is unstructured and in natural language. Thus, semantic networks [26] (i.e., directed graph where nodes represent entities or concepts, and the arcs mark relationships between them) are preferred. Semantic networks provide a better contextualization in terms of the representation and identification of knowledge facts. Knowledge graphs are semantic networks which include inference methods for deriving and combining information in order to extract new facts. NoSQL graph databases include different query languages to read or modify stored data [1]. The most relevant examples of such query languages are Cypher [13] (which introduces powerful pattern-matching based queries), SPARQL [23] (designed for RDF data storage schemas), GraphQL (a data query language for APIs) and Gremlin [25] (which displays a distinguishable functional style in its queries).

Different approaches for structuring information in graph databases were considered for representing knowledge in our conversational agent [22]. One of them, the labeled-property graph model (used, for example, by the Neo4j Database Management System [20]) consists of adding properties to nodes or relationships to specify different metadata in a key-value fashion. This approach enables a flexible and compact representation of information. Conversely, the RDF model [18] allows nodes and arcs only to specify an URI label, which in return defines the class of the entity or a relationship. This leads to a more strict and uniform data representation, but, at the same time, to a reduced topological conciseness. Additional properties have to be injected as separate nodes due to the simplicity of the RDF [18] graph elements.

METHOD

Corpus

Two supervised learning models are involved in the Natural Language Understanding pipeline of our agent: the first for detecting the intent from the user's utterances, and the second for parsing the sentences. These models implied the creation of two datasets containing labeled examples.

Intent Classification

The classification of the user's sentences by their scope requires defining a set of intents that determines the general types of tasks handled by the agent. These intents outline the constrained world or domain the agent was designed for. The considered intent classes are presented in Table 1.

The training dataset consists of 285 manually built sentences (for the "get" and "store" intents) and phrases (for auxiliary intents, such as greetings or the store request announcement). Additionally, a separate test set comprising of 128 examples was created to evaluate the performance of the intent detector. The distribution of examples in the training and test datasets can be observed in Figure 2.

Syntactic Parsing

Due to lack of a specific dataset for our task in Romanian, the training dataset consists of 200 manually annotated examples, while the test set used to measure parsing accuracy contains another 70 sentences. The examples were created using a custom-made HTML and JavaScript web interface that led to a faster visual annotation process. **Error! Reference source not found.**Figure 3 shows the frequencies for each syntactic question introduced in the training set. An example with an annotated sentence can be observed in Figure 4. The head represents the index of the word in the sentence to which another word is connected to in the dependency tree; the „deps" property includes custom word tags used later on.

Table 1. Intents description.

Intent	Description
Greet, Goodbye	Can be used at the beginning or the end of an interaction with the agent, although their use is optional. General greeting phrases were used for these intents, as training examples.
Store request	The user announces the agent that they want to store a piece of information in the knowledge graph.
Store/Get location	The user wants to tell/ask the agent about the location of an entity or an event. For location requests, the agent expects a sentence consisting of the interrogative pronoun "unde?" (eng., "where?"), along with a subject entity or an action involving a direct object. To store a location, a sentence containing a place adverbial and either a subject (e.g., tell where the subject is placed) or a direct object (e.g., tell where the user puts an object) is expected by the agent.
Store/Get timestamp	The sentence tells/asks the agent about the timestamp (time point, duration, or frequency) of an event. The phrases "când?" (eng., "when?"), "de când?" (eng., "since when?"), "până când?" (eng., "until when?") and "cât timp?" (eng., "how long?") are used to ask the agent for this type of information, along with an event expressed as a bare noun phrase or as a complex action (containing also other semantic entities such as adverbials or direct objects). A sentence containing a time adverbial is expected to register a time information.
Store/Get attribute	The user wants to store or retrieve the value of an entity attribute (a personal detail of that entity, which can range from a person's phone number to the dimensions or the price of an object). The "get attribute" intent is represented, in the agent's view, through a question involving a phrase of the type "care (a fi)?" (eng., "what is?"), followed by a noun phrase describing the requested property (and optionally its owner). To store an attribute, a sentence containing a subject (the entity) and a predicate (specifying the attribute value) is expected by the agent.
Store following attribute	Similar to the previous action, this intent allows the user to store an attribute, but in two steps: the first utterance is matched to this intent category and contains the attribute name, whilst the second utterance includes the actual attribute value. This approach enables users to store complex values that should be taken as a whole, instead of being parsed into component tokens.
Get subject	The user asks for the entity that represents the subject of an action or has a specific property according to the knowledge base. This intent is triggered through the interrogative pronoun "cine?" (eng., "who?"), alongside an action or a state.

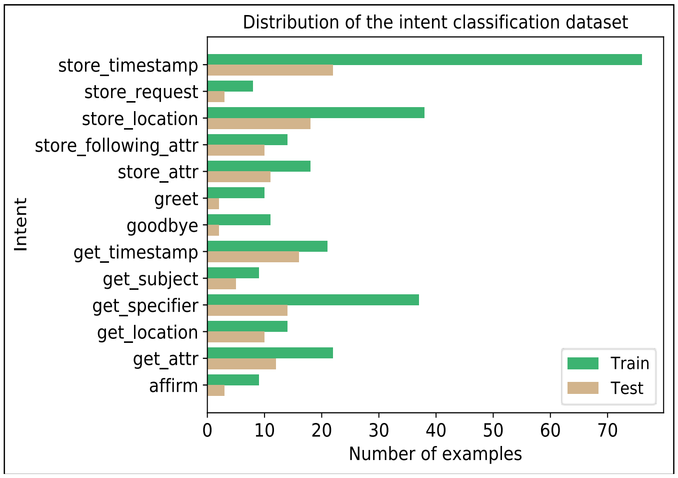


Figure 2. Distribution of train and test examples for intent classification.

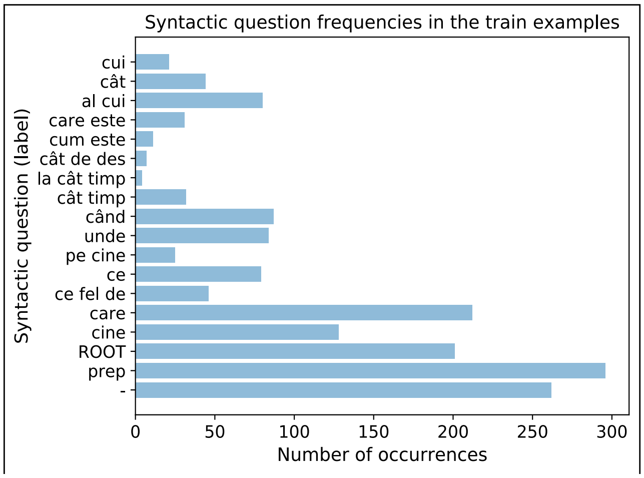
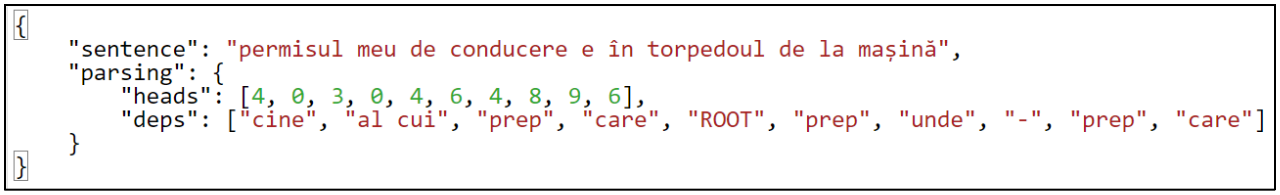


Figure 3. Frequencies of various syntactic questions in the training examples.



```
{
    "sentence": "permisul meu de conducere e în torpedoul de la mașină",
    "parsing": {
        "heads": [4, 0, 3, 0, 4, 6, 4, 8, 9, 6],
        "deps": ["cine", "al cui", "prep", "care", "ROOT", "prep", "unde", "-", "prep", "care"]
    }
}
```

Figure 4. Annotated sentence example.

Application Architecture

The application is divided into several components that perform sequential processing tasks, starting from the user's utterances (speech or text), through the NLU (Natural Language Understanding) pipeline, to the knowledge base, and back to the user with responses to their information requests, as depicted in Figure 5.

Processing Pipeline

Our conversation agent is built on top of the RASA framework for conversational agents [5], which provides two chained pipelines: the first for NLU, while the second (RASA Core) is centered on dialogue management (i.e., selecting the proper response to the user's input and advancing the conversational flow).

The modularity of the framework allows the customization of the pipeline components; thus, an application-specific processing pipeline was designed. The first components are responsible for tokenization and feature generation. They are based on the spaCy [14] Romanian model (available in the open-source ReaderBench framework [10]), which is trained on the Romanian Universal Dependencies treebank [4]. This approach considers pretrained word embeddings, which offer additional linguistic features that support the next NLU components (such as the intent classifier) to obtain better performance. In addition to the previous components, other featurizers predefined in RASA, namely the Lexical-Syntactic Featurizer and the Count Vectors Featurizer were added. The latter was configured to create the bag-of-words representation only at token level because the character and the n-gram levels were generating too much variance in the results. Finally, a custom component for syntactic parsing was integrated in the NLU pipeline, based on a pretrained model and a set of custom actions provided to the RASA Core pipeline to be executed (depending on the response selection result) as a reply to the user's request. These actions are responsible for establishing a connection to the database, as well as returning the results to the user or performing the database update.

User Interface

The User Interface (UI) of our conversational agent consists of a single-page web application (HTML, CSS and JavaScript) based on the React framework [24] (see Figure 7). The interface consists of a chat window in which the exchange of messages conducted with the agent in the current session are displayed. The UI has a multimodal input as the user can interact with the bot either by typing the requests in an input text field from the chat window, or by uttering a statement after launching the speech input mode.

Speech-to-text conversion is performed through the Web Speech API [9]. It is a JavaScript API, currently specified as a W3C draft, independent of the underlying algorithm implementation that provides a unified interface for performing both automatic speech recognition and speech synthesis in the context of web applications.

Communication with the core agent is based on the RASA's HTTP REST API, which is used to deliver the user's messages to the bot, and then wait for its response. The requests are executed asynchronously using the Axios HTTP client [3], which ensures a fluid interaction between the user and the agent.

The complete communication between the modules of the application can be observed in Figure 6. Note that, although custom actions are also part of the RASA framework, these services run as a separate server which is accessed from the core agent through an HTTP endpoint.

The application follows a modular MVC (Model-View-Controller) architecture [16], where the view is represented by the web interface, the custom actions together with the Neo4j database define the data model, whereas the core RASA agent acts as a controller. The RASA agent receives the user's requests from the view, processes them, and sends the results to the model component, which manages data representation. Conversely, the controller builds the response for the user based on the extracted data and delivers it by means of the frontend view, which presents it to the user.

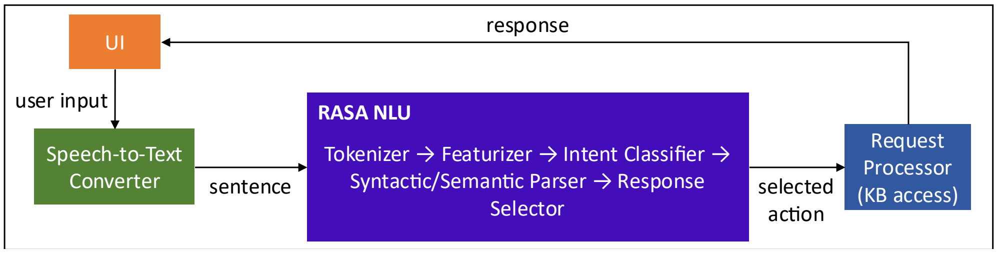

UI
response
user input
Speech-to-Text Converter
sentence
RASA NLU
Tokenizer → Featurizer → Intent Classifier → Syntactic/Semantic Parser → Response Selector
selected action
Request Processor (KB access)

Figure 5. Logical flow diagram of the application.

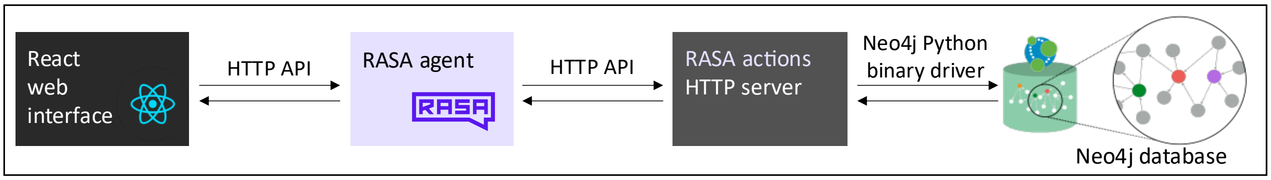
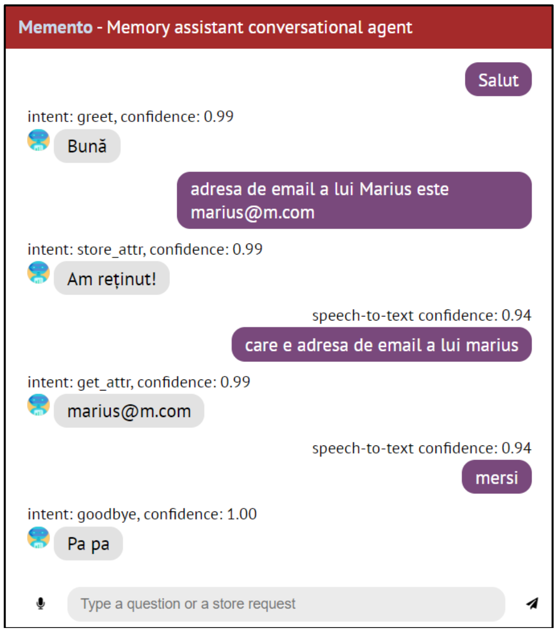


Figure 6. Main application modules.



Memento - Memory assistant conversational agent

Salut

intent: greet, confidence: 0.99

Bună

adresa de email a lui Marius este marius@m.com

intent: store_attr, confidence: 0.99

Am reținut!

speech-to-text confidence: 0.94

care e adresa de email a lui marius

intent: get_attr, confidence: 0.99

marius@m.com

speech-to-text confidence: 0.94

mersi

intent: goodbye, confidence: 1.00

Pa pa

Figure 7. Application interface.

Intent Classifier

Our agent maps the user sentences to a limited set of predefined intents, in accordance with Table 1. The intent classifier allows the request processor to perform a more customized entity selection, depending on the relevant information included in the input phrase. In addition, the classifier gives the agent the ability to execute an action specific to an identified intent, i.e. run a query or update the knowledge graph. A noteworthy aspect is that a single intent may be selected for an utterance, meaning that no more than one kind of relevant information (specific to that intent) is identified and considered by the agent.

Intent classification is performed using RASA's DIET classifier. Corresponding part of speech tags for each word were provided to the classifier by customizing RASA's Lexical-Syntactic Featurizer in order to accommodate the particularities of statements specific to each intent. Additionally, the RASA entities occurring in the examples of each type of intent were declared. The presence or the absence of such entities in an example can be used, according to the RASA documentation, as features by the DIET classifier to improve its accuracy.

Request Processor

After the intent from the user statement is predicted and the semantic entities are extracted, the processing pipeline continues by executing an action specific to a given intent. The action is selected based on the stories provided as training data to the RASA Core pipeline. The stories consist in general of chains of pairs (intent, reply action), describing possible conversational flows. However, since the majority of our intents imply only a request-response exchange, the associated stories only include the mapping between the intent and the custom action that accesses the knowledge base to process the user's inquiry. Additionally, an acknowledgement message follows when storing new information to confirm the action's fulfillment, so that the user knows the information was registered. A slightly different case is the one of the "store following attribute" intent because the attribute name needs to be retained between two replies. This is achieved by means of a RASA form, which is initiated once the first user's message (the store request) is uttered. At that point, a slot is filled with the attribute description. The form is completed after the user inserts the value of the attribute as the second reply. Afterwards, the action responsible for storing the attribute is executed.

Knowledge Representation

Our knowledge base is based on the Neo4j graph database management system, which ensures a powerful semantic representation of the entities and the facts associated to them. A notable benefit is the lack of need for an explicit database schema, meaning that each entity may have its own types of attributes or relationships with other entities. This facilitates the storage of heterogeneous information transmitted by the user. Another reason for choosing Neo4j is Cypher, its query language [13], which enables pattern matching queries. Hence topological structures such as nodes, relationships, paths or subgraphs can be matched using a pattern that intrinsically specifies labels or properties of the nodes and the relations that link them. The reason for selecting Neo4j resides in its ease of integration with RASA. Currently, there are 4 alternative solutions that can be used [7]. Our differentiating criterion between them consisted of publicly available performance benchmarks [11].

Entities defined as noun phrases are represented as a tree in the graph, where each node identifies a syntactic component of the noun phrase. The decomposition is performed in the following manner:

- The root (noun) of the noun phrase constitutes the base class of the entity.
- Each span with the syntactic function of the attribute, responding to one of the questions "care?" (eng., "which?") or "ce fel de?" (eng., "what kind of?") represents a "specifier", having the role of identifying a particular instance of the base class.
- The token responding to the question "al cui?" (eng., "whom") identifies the owner of the entity, and therefore it is placed as the root of the entity's tree representation. This token can be also classified as a "specifier".

RESULTS

The customized intent classifier from RASA exhibited very good results at identifying the intents from the test sentences, as shown in the confusion matrix from Figure 8 where all test inputs were correctly classified. Confidence was around 95% on all entries from the test dataset, denoting a strong differentiation between intents.

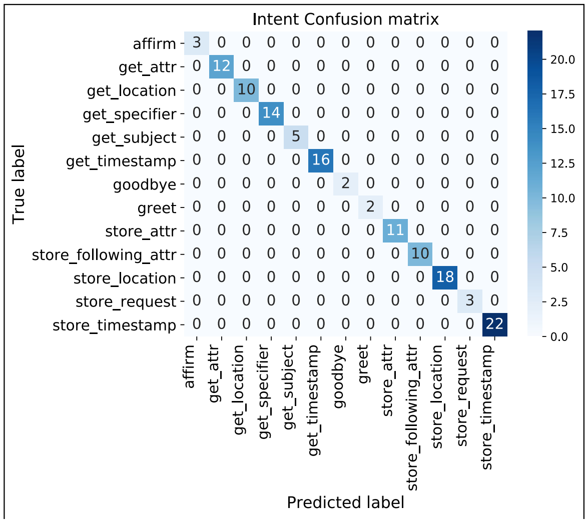


Figure 8. Confusion matrix for the intent classification task.

The accuracy of the syntactic parser is high (93% for dependency heads). Figure 9 depicts the confusion matrix resulted after running the parser on our test dataset consisting of 70 examples. Syntactic dependencies can occur in different structures and positions within a sentence; thus, a larger variety in the training dataset is required to ensure the model's capability to generalize. However, our current set of examples, which were manually created and annotated, is quite limited and needs to be extended.

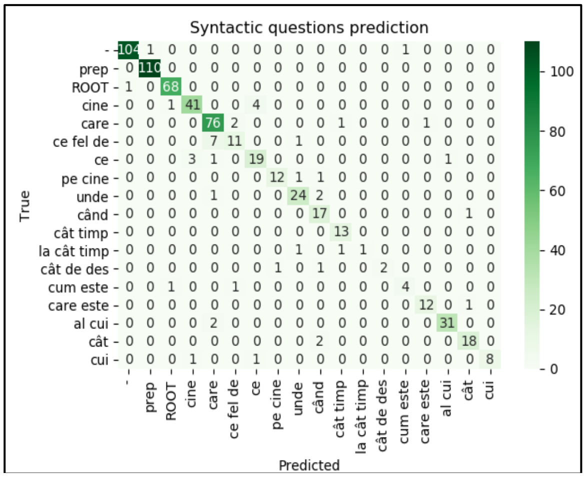


Figure 9. Confusion matrix for the syntactic parsing.

Problems also occurred when the input sentence included tokens not belonging to any of the available syntactic dependency classes. Nevertheless, the classifier tries to map these tokens to a class, which in many cases was not the "-" dependency (i.e., the best option in this case). This results in wrong labels, and may also lead to words around the misclassified tokens being mapped to an incorrect dependency. Table 2 includes the precision, recall, and F1 scores obtained for each type of syntactic question. Only the labels' attachment is considered, without taking into account the placement of the dependency heads.

Table 2. Syntactic questions classification.

Type	P	R	F1-score	#
-	0.99	0.98	0 .99	106
ROOT	0.97	0.99	0.98	69
al cui (whose)	0.97	0.94	0.95	33
care (which)	0.87	0.95	0.91	80
care este (which is)	0.92	0.92	0.92	13
ce (what)	0.79	0.79	0.79	24
ce fel de (what kind of)	0.79	0.58	0.67	19
cine (who)	0.91	0.89	0.90	46
cui (to whom)	1.00	0.80	0.89	10
cum este (how is it)	0.80	0.67	0.73	6
când (when)	0.74	0.94	0.83	18
cât (how much)	0.90	0.90	0.90	20
cât de des (how often)	1.00	0.50	0.67	4
cât timp (how much time)	0.87	1.00	0.93	13
la cât timp (how long)	1.00	0.33	0.50	3
pe cine (who)	0.92	0.86	0.89	14
unde (where)	0.89	0.89	0.89	27
preposition	0.99	1.00	1.00	110
Accuracy			0.93	615
Macro avg	0.91	0.83	0.85	615
Weighted avg	0.93	0.93	0.93	615

CONCLUSIONS

In this article we introduced a chatbot for Romanian capable of storing and retrieving information from and to users. Several components were designed, namely a web interface, an intent classifier, a syntactic parser, custom actions dedicated to each intent, and a graph database manager. The user interface consists of a React web page that mediates the communication between the client and the conversational agent. Speech-to-text capabilities were added to facilitate the user interaction by using the Web Speech API.

As for future improvements, we consider a smarter data matching algorithm to identify the requested information, even if the request is not completely similar to the stored information (for example, words replaced with synonyms, or missing prepositions linking tokens from a noun phrase). In addition, we want to integrate additional information from general knowledge graphs (e.g., DBPedia [2]). This would allow the agent to match entities, states, or actions in a more generalized manner, resulting in a more intelligent and helpful behavior. Another future step to ensure increased performance and robustness would consist of enhancing the syntactic parser component. We strive to handle any type of statements, including those containing subordinate sentences, with all potential types of syntactic dependencies.

ACKNOWLEDGMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 72PCCDI / 2018, ROBIN – "Roboții și Societatea: Sisteme Cognitive pentru Roboți Personali și Vehicule Autonome".

REFERENCES

1. Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., and Vrgoč, D., 2017. Foundations of modern query languages for graph databases. *ACM Computing Surveys (CSUR) 50*, 5, 1–40.
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z., 2007. DBPedia: A nucleus for a web of open data. In *The semantic web* Springer, 722–735.
3. Axios, n.d. Axios homepage. Retrieved July 27th 2020 from https://github.com/axios/axios.
4. Barbu Mititelu, V., Ion, R., Simionescu, R., Irimia, E., and Perez, C.-A., 2016. The Romanian Treebank Annotated According to Universal Dependencies. In Proceedings of the 10th Int. Conf. on Natural Language Processing (HrTAL2016) (Dubrovnik, Croatia).
5. Bocklisch, T., Faulkner, J., Pawlowski, N., and Nichol, A., 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.
6. Brandtzaeg, P.B. and Følstad, A., 2017. Why people use chatbots. In Proceedings of the International Conference on Internet Science (Thessaloniki, Greece), Springer, 377–392.
7. Bunk, T., 2019. Set up a knowledge base to encode domain knowledge for Rasa. Retrieved September 27th 2020 from https://blog.rasa.com/set-up-a-knowledge-base-to-encode-domain-knowledge-for-rasa/.
8. Bunk, T., Varshneya, D., Vlasov, V., and Nichol, A., 2020. DIET: Lightweight Language Understanding for Dialogue Systems. *arXiv preprint arXiv:2004.09936*.
9. Community Group Report, 2020. Web Speech API. Retrieved July 27th 2020 from https://wicg.github.io/speech-api/.
10. Dascalu, M., Dessus, P., Bianco, M., Trausan-Matu, S., and Nardy, A., 2014. Mining texts, learner productions and strategies with ReaderBench. In *Educational Data Mining: Applications and Trends*, A. Peña-Ayala Ed. Springer, Cham, Switzerland, 345–377.
11. DB-Engines, 2020. DB-Engines Ranking of Graph DBMS. Retrieved September 27th 2020 from https://db-engines.com/en/ranking/graph+dbms.
12. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
13. Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Selmer, P., and Taylor, A., 2018. Cypher: An evolving query language for property graphs. In Proceedings of the 2018 Int. Conf. on Management of Data, 1433-1445.
14. Honnibal, M. and Montani, I., 2017. spacy 2: Natural language understanding with bloom embeddings. *convolutional neural networks and incremental parsing 7*, 1.
15. Jurafsky, D. and Martin, J.H., 2009. *An introduction to Natural Language Processing. Computational linguistics, and speech recognition*. Pearson Prentice Hall, London.
16. Krasner, G.E. and Pope, S.T., 1988. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming 1*, 3, 26–49.
17. Lafferty, J., McCallum, A., and Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th Int. Conf. on Machine Learning 2001 (ICML 2001) (Williamstown, MA, USA), ACM, 282–289.
18. Lassila, O. and Swick, R.R., 1998. *Resource description framework (RDF) model and syntax specification.* World Wide Web Consortium.
19. Liu, B. and Lane, I., 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.
20. Miller, J.J., 2013. Graph database applications and concepts with Neo4j. In Proceedings of the Southern Association for Information Systems Conference (Atlanta, GA, USA).
21. Pennington, J., Socher, R., and Manning, C.D., 2014. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP 2014) (Doha, Qatar), ACL.
22. Pokorný, J., 2015. Graph databases: their power and limitations. In Proceedings of the IFIP Int. Conf. on Computer Information Systems and Industrial ManagementSpringer, 58–69.
23. Prud'hommeaux, E. and Seaborne, A., 2017. SPARQL query language for RDF. W3C Recommendation (2008) World Wide Web Consortium.
24. React, n.d. React website. Retrieved July 27th 2020 from https://reactjs.org/.
25. Rodriguez, M.A., 2015. The Gremlin graph traversal machine and language (invited talk). In Proceedings of the 15th Symposium on Database Programming Languages (Pittsburgh, PA, USA), 1–10.
26. Sowa, J.F., 2014. Principles of semantic networks: Explorations in the representation of knowledge Morgan Kaufmann, San Mateo, CA, USA.
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I., 2017. Attention is all you need. In Proceedings of the 31st Conf. on Neural Information Processing Systems (NIPS 2017) (Long Beach, CA, USA), 5998–6008.
28. Wang, Y., Shen, Y., and Jin, H., 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. *arXiv preprint arXiv:1812.10235*.

Seeking an Empathy-abled Conversational Agent

Andreea Grosuleac
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
andreea.grosuleac@stud.acs.upb.ro

Ştefania Budulan
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
stefania.budulan@cs.pub.ro

Traian Rebedea
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
traian.rebedea@cs.pub.ro

DOI: 10.37789/rochi.2020.1.1.16

ABSTRACT

A fairly novel area of research, at the conjunction between Artificial Intelligence (AI) and Human-Computer Interaction (HCI), resides in developing conversational agents as more users prefer this type of interaction to conventional interfaces. In this paper, we present an open-domain empathic chatbot, encompassing two of the biggest challenges of dialog systems: understanding emotions and offering appropriate responses. Although these tasks are trivial for a human, it is difficult to create a system that can recognize others' feelings in a discussion. The proposed model is developed based on the Generative Pre-Trained Transformer and it uses two datasets, PersonaChat and Empathetic Dialogues to achieve an empathic chatbot with a cordial personality. The measured performance - 18.20 perplexity, 6.56 BLEU score, and 6.56 accuracy - comes close to the state-of-the-art models, while offering a further refined dialogue persona.

Author Keywords

Empathic dialogue; Open-domain conversational agents; Empathetic chatbot; Conversational interfaces.

INTRODUCTION

As observed by Følstad and Brandtzæg [4] nowadays the interaction between humans and computers seems to turn toward natural-language user interfaces making the development of conversational agents a vital area of research in HCI.

Empathic chatbots are conversational agents that are not only able to generate emotional responses, but can understand the feelings of a user and respond accordingly. While prior work focused on creating conversational systems that can speak coherently and grammatically correct, in the last years the attention of the academic community changed to a more engaging agent that can mimic a real person's skills [14].

This paper tackles the problem of empathic chatbots, a new research focus, that tries to address the lack of empathy in the widely available conversational agents. This is an important issue as there is a current exponential growth in the spread of such agents to solve mundane tasks as website guidance, entertainment, information extraction, or question answering in domains like customer service or education. An important stand-alone application for an empathic chatbot can be made in the healthcare domain as more young people report a decreasing number of friends and personal connections, proving that our generation may be the most connected one, yet the loneliest.

The current work proposes an empathic chat agent that embodies a personality that provides the ability to create more engaging and natural responses. To recognize feelings during the conversation we train a classifier that can predict emotions from a context offered by the user and a small history of the conversation using fastText [8]. To generate responses we apply fine-tuning of a generative pre-trained model [12] using PersonaChat [19] and Empathetic Dialogues [15] datasets.

RELATED WORK

Rashkin et al. [15] proposed a new benchmark for empathic dialogues that is a necessity for further research on this topic. In their work, they designed a novel dataset and tested two types of architectures using this data: retrieval-based, using BERT encoder [2] to find the best match, and generative, using Transformers [15].

MoEL [9] proposed a system that softly combines responses from multiple empathic listeners for each emotion. Despite the occasional confusion created by trying to generate a response from a high variance emotional distribution, the model achieves better results than a generic multi task transformer [17].

CAiRE [10] is the most recent empathic chatbot and it achieves state-of-the-art performance. PersonaChat [19] and Empathetic Dialogues [15] datasets are used to create an open-domain, end-to-end conversational agent.

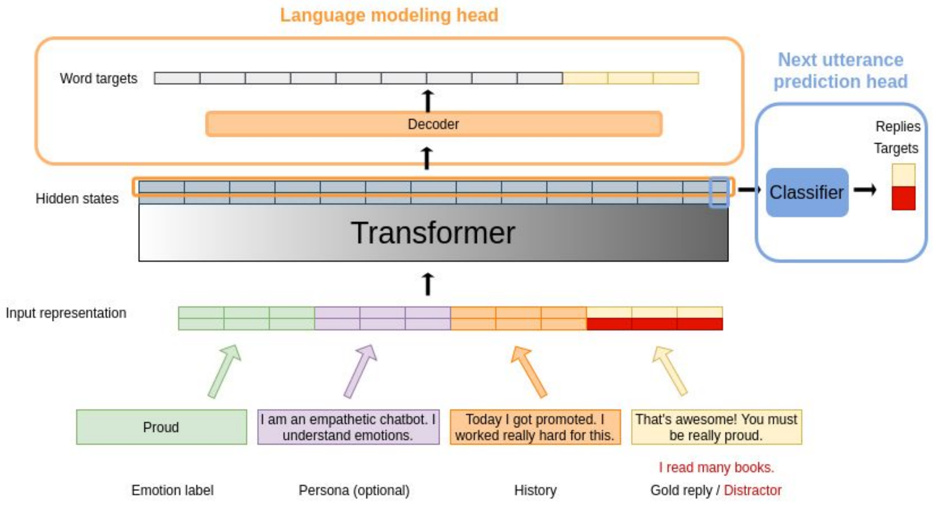


Figure 1. The model architecture used by the proposed empathic agent.

A recent work that focuses on creating a human-like conversational agent is presented by Roller et al. [16]. The authors aim to obtain a recipe for a robust open-domain chatbot that presents not only empathy, but also personal background and knowledge.

Datasets

PersonaChat [19] dataset makes a step towards a more engaging dialogue. This is a very useful dataset because it proposes a novel turn-based dialogue that can improve a conversational agent with a consistent personality. This dataset is made over 160,000 utterances between crowdworkers from Amazon Mechanical Turk, who received a different persona to mimic during the conversation. Each dialogue has a minimum 6 or 8 utterances, each with a limit of 15 words.

Empathetic Dialogues [15] is a novel dataset based on sentimental conversations between a speaker, who describes a situation when they felt in a certain way, and a listener that has to respond empathically. This dataset provides 32 emotions distributed almost evenly over 24,850 conversations from 810 Amazon Mechanical Turk workers. Each pair was asked to choose an emotion and every participant had to provide a description of a scene when they felt that way.

PROPOSED SOLUTION

The main steps followed by our solution are preprocessing the data, transforming the data to match with the pre-trained model input representation, converting the inputs into tensors, and finally, fine-tuning the model as presented in Figure 1. For the emotion classifier, this work uses an external trained model for text classification, fastText [3].

The base architecture we used in our work is a unidirectional GPT (Generative Pre-trained Transformer). The model follows standard fine-tuning [18] and it is based on the open-source implementation published on GitHub [5] which was extended for the Empathetic Dialogues dataset.

Input Representation

The GPT model receives as input a sequence of words. To fine-tune this model for a dialogue task, all the main features of a conversation must be represented in the resulting embedding space. The current work follows the same design presented by Wolf et al. [18] for the PersonaChat dataset, adapted to the Empathetic Dialogues.

To speed up training, other than the gold reply, represented by the correct next utterance, one distractor was used. The distractor was picked randomly from a pool of candidates that includes all possible individual utterances from the training dataset.

The input fed to the network must contain the concatenation of the emotion label, the custom persona (”i am an empathetic chatbot.”, ”i understand emotions.”, ”i am friendly.”, ”i want to help humans.”), the history of the conversation and the correct reply or the distractor. Apart from the word embeddings, the model needs more information about the input, such as the position for each token used by the attention mechanism and delimitation between the segments.

The delimitation is made using special tokens such as the start of a sequence, the end of a sequence and the indexes of the two speakers. A padding token is used to fill the remaining positions up to 512, which represents the length of the input sequence.

Training

In our work, we start from the GPT Double Heads [6] model and the corresponding tokenizer [7] released by OpenAI. We use a Double Heads model because during the fine-tuning we optimize a multi-task loss function that combines both language model loss and next-sentence prediction loss. For the transfer learning part, there are used 5 epochs, a batch size of 4, a learning rate equal to *6.25e-5* that was linearly decayed to zero during training, a max history for each speaker of 2 and 4 steps of gradient accumulation. The current work uses AdamW, an improved version of Adam optimizer.

To train the classifier in order to predict the emotion, the current work utilizes the fastText [3] library. The text from which the model learns is assembled from the context and the full conversation. The model is trained for 50 epochs and with a learning rate equal to *0.7*.

Decoding

During inference, a decoder is used to predict the next utterance as a sequence of words, based on the current input. The algorithm used is the combination of the top-k and top-p sampling. Top-*k* sampling reduces the candidates at the best *k* possibilities with the highest probability and top-*p* sampling keeps only the candidates for which the sum is greater than the *p* parameter. Therefore there are kept only the tokens with the higher probability acquired from the two different methods.

The parameters used for decoding are *k=0.8* and *p=0.9*. Other parameters used are *max_length=20* that limits the number of generated words and *min_length=1*. The max number of utterances in the history is 4.

EXPERIMENTS AND RESULTS

In this section, there will be presented two experiments that aim to explore different possible approaches and to compare their effectiveness in the context of dialogue systems that show empathy. The evaluation metrics used are perplexity and BLEU score to assess the current proposal performance relative to other models used for the same goal, and accuracy to measure the emotion classification results. Perplexity refers to the uncertainty from the model prediction and it measures how well it predicts the next golden token for the test data and BLEU score measures the quality of a text by calculating the distance between the generated text and the golden reply.

The first experiment is fine-tuning the GPT network using the Empathetic Dialogues dataset (denoted as **GPT+ED**). In this case, the input is built using the emotion label, the history, and the reply.

The second experiment is using transfer learning from the Empathetic Dialogues to the pre-trained GPT model using PersonaChat (denoted as **GPT+PC+ED**). For this attempt, we started from a pre-trained model [11]. In this case, the model already performs well on a general dialogue task and the experiment aims to adapt it for a more empathic version by prepending the emotion at the top of the input.

In Table 1 are presented the scores obtained by our models. Both of them are evaluated on the Empathetic Dialogues dataset. The best results in the domain of empathic chatbots are currently obtained by CAiRE [10], the state of the art model. The improvements that CAiRE brings are the addition of a persona that supports the empathic component as well as using a pre-trained model such as GPT. These aspects, that differentiated CAiRE from other models, were followed by the current work and the results presented in Table 1 show that the **GPT+PC+ED** version has only slightly lower scores.

Table 1. Comparison between existing models in the domain of empathic chatbots and the current work. The first five models were introduced by Rashkin et al. [15], based on a Transformer, pretrained on 1.7 billion conversations from Reddit. (The first six scores are taken from [15] and [10])

Model	PPL	BLEU	Accuracy
Pretrained	27.96	5.01	-
Fine-Tuned	21.24	6.27	-
MULTITASK	24.07	5.42	-
EmoPrepend-1	24.30	4.36	-
ENSEM-DM	19.05	6.83	-
CAiRE	**13.32**	**7.03**	**0.516**
GPT+ED	21.34	5.82	0.483
GPT+PC+ED	*18.20*	*6.56*	*0.483*

Table 2. Examples of generated answers and the predicted emotion based on an initial context.

Context	Real / Predicted	GPT+ED	GPT+PC+ED
I am so impressed by my current boss. He is a terrific example of leadership in every way.	Impressed / Impressed	that is awesome! i bet you are so happy to have him as your boss.	that is good to hear. what makes you think that?
I recently got very upset with my sister when said some very mean things to me. I let her know that she was out of line by yelling at her. I	Angry / Guilty	it's okay. you have to be honest with her.	i'm sorry, what did she say?

feel bad for both of us.			
I hate it when my husband goes away for work. There is no one else here but me.	Lonely / Lonely	why do you hate it when he is gone?	that's a bummer. have you tried talking to him about it?

One main difference perceived between the two experiments is the type of sentence the model generates as shown in Table 2. **GPT+ED** tends to generate statements that interrupt the flow of the conversation and that focus just on being empathic towards the speaker. **GPT+PC+ED** usually contributes to the discussion by asking questions while still revealing empathy. This dissimilarity may be caused by the datasets the models use. While Empathetic Dialogues focuses on expressing the emotion recognized from the text, the PersonaChat dataset has conversations where speakers want to find out more about each other.

By analyzing the confusion matrix for the emotion classifier, it shows that it performs well and predicts in most of the cases the correct label. For some sentiments such as “angry” and “furious” or “terrified” and “afraid”, the classifier is not capable of distinguishing properly, as they are close in signification.

Failure cases

Some of the failure cases met during the evaluation process consist in repetition of words and ideas, the inability to refer to past information or shallow understanding of concepts and ideas. However, these are recurring problems in conversational agents and may require fundamental novel approaches to mitigate the occasional poor performances.

CONCLUSION

In this work, we introduced an empathic conversational agent that can understand emotions during a discussion and respond properly. During the development of the dialogue system, we provided two experiments that show that the best approach is to endow the chatbot with a friendly personality that can help it generate more engaging and empathic answers. Using the PersonaChat dataset for fine-tuning the model increases the chit-chat capability and entertains longer conversations.

The main contribution of our paper is the proposal to not integrate the emotion classification into the training pipeline, and to independently learn to classify emotions to ease the training process and to make this prediction more robust. One of the immediate improvements that can greatly improve the performance of the model is to replace the transformer architecture with improved versions of GPT, such as GPT 2 [13] or GPT 3 [1].

The current results are promising and with further improvements, we will be able to build a truly empathic chatbot that displays more skills from humans’ social behavior and that can be used at large scale to nourish the emotional and social needs of people.

REFERENCES

1. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Agarwal, S. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
3. FastText library https://fasttext.cc/ last accessed on 22 June 2020.
4. Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. Interactions, 24(4), 38-42.
5. GitHub https://github.com/huggingface/transfer-learning-conv-ai last accessed on 8 July 2020.
6. GPT Double Head model https://huggingface.co/transformers/model_doc/gpt.html#openaigptdoubleheadsmodel last accessed on 8 July 2020.
7. GPT Tokanizer https://huggingface.co/transformers/model_doc/gpt.html#openaigpttokenizer last accessed at 8 July 2020.
8. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.
9. Lin, Z., Madotto, A., Shin, J., Xu, P., & Fung, P. (2019). Moel: Mixture of empathetic listeners. arXiv preprint arXiv:1908.07687.
10. Lin, Z., Xu, P., Winata, G. I., Siddique, F. B., Liu, Z., Shin, J., & Fung, P. (2020). CAiRE: An End-to-End Empathetic Chatbot. In AAAI (pp. 13622-13623).
11. Pretrained GPT with PersonaChat https://s3.amazonaws.com/models.huggingface.co/transfer-learning-chatbot/gpt_personachat_cache.tar.gz last accessed at 17 June 2020.
12. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.
13. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.
14. Radziwill, N. M., & Benton, M. C. (2017). Evaluating quality of chatbots and intelligent conversational agents. arXiv preprint arXiv:1704.04579.
15. Rashkin, H., Smith, E. M., Li, M., & Boureau, Y. L. (2018). Towards empathetic open-domain conversation models: A new benchmark and dataset. arXiv preprint arXiv:1811.00207.

16. Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., ... & Boureau, Y. L. (2020). Recipes for building an open-domain chatbot. arXiv preprint arXiv:2004.13637.
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
18. Wolf, T., Sanh, V., Chaumond, J., & Delangue, C. (2019). Transfertransfo: A transfer learning approach for neural network based conversational agents. arXiv preprint arXiv:1901.08149.
19. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., & Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too?. arXiv preprint arXiv:1801.07243.

Analysis of convergence and divergence in chat conversations

Liviu-Andrei Niță

University Politehnica of Bucharest
312 Splaiul Independenței,
Bucharest, Romania
nitaliviu79@yahoo.com

Ștefan Trăușan-Matu

University Politehnica of Bucharest
312 Splaiul Independenței,
Bucharest, Romania
and
Research Institute for Artificial Intelligence
and
Academy of Romanian Scientists
stefan.trausan@upb.ro

Traian Rebedea

University Politehnica of Bucharest
312 Splaiul Independenței,
Bucharest, Romania
traian.rebedea@upb.ro

DOI: 10.37789/rochi.2020.1.1.17

ABSTRACT

A discussion between several participants is often accompanied by an exchange of information between speakers and, in the case of collaborative learning, a polyphonic interanimation is desired. They have, according to the polyphonic model, different points of view that are convergent or divergent by which one of the participants approves or disapproves of another person that is participating in the discussion. This paper presents an approach for identifying convergent or divergent utterances using neural networks and other machine learning methods in order to help people analyze a dialogue conducted in an online environment. Especially in collaborative chats used in education, this solution allows professors to identify how the participants in the discussion exchange information and how the debate evolved in time.

KEYWORDS: dialogue, chat, neural networks, machine learning, convergent, divergent, polyphony.

ACM Classification Keywords

I 2.7 Natural Language Processing. Text analysis.

INTRODUCTION

Online conversations are exchanges of ideas in synchronous or asynchronous ways between two or more participants. A pattern observed in a significant number of discussions is the accompaniment of the exchange of information between individuals by different points of view, which are either convergent or divergent, in which a sender approves or disapproves the ideas of another person participating in the discussion. This phenomenon may be viewed as an interanimation between ideas, similarly to the interanimation of voices in polyphonic music [1,2]:

> “a suggestion for how to analyse knowledge building and debates of ideas is to consider the complex weaving of musical polyphony, where several voices/ideas are concurrently developed in time, with the consequence that some dissonances/divergences occur, but eventually a coherent whole, a discourse, is achieved.” [1]

We consider that an utterance in a conversation has a convergent character when put in relation with a message send by another participant in the discussion expresses an agreement of opinions between individuals. The process by which a convergent statement is formed is that of extracting, defining, and emphasizing the information transmitted by the original sender [3]. For example, if a person says (the examples are taken from real chats between students [4], therefore they may contain mistypes and other linguistic errors):

> *“I think that blog are there to keep in touch with the sensitive part of the customer”*

and another participant responds with the message:

> *“I support you on that”*

between these replies it has been established a point of convergence.

Unlike convergent statements, divergent remarks are defined by a process that generates different points of view to reach a common truth that has the effect to stimulate creativity [5,6]. For example, when a participant in a chat says:

> *"I think that blog are there to keep in touch with the sensitive part of the customer…but need a lot of men power which can be a big drawback for small companies"*

and another one re:

> *"a small company would need no more than two or three specialists to cope with their customers in a live chat systems"*

then we can say that a divergence exists between the two utterances.

These two types of replies are the basic components of interanimation that characterizes polyphony, a desired phenomenon arising in collaborative chats. By identifying them, different performance indicators for collaborative learning processes may be provided by learning analytics applications [1] allowing teachers to observe, develop new strategies to increase student performance, and ultimately achieving better educational dialogue by developing the ability of students to accept different points of view and increase the level of knowledge that is transmitted in chat conversations and in general [1,8].

STATE OF THE ART

The basic structural unit of a dialogue is the utterance (reply). In the format that dialogue is a conversation between at least two people who alternate the roles of speakers and listeners, the utterance becomes the element that transmits a quantity of information. The analysis of a conversation implies the understanding of the effect of the messages transmitted through a reply by studying the (polyphonic) structure of the dialogue, the action sequences, and the adjacent pairs within a dialogue. An adjacent pair consists of a sequence of two sentences produced by different participants in the discussion in which a first action transmitted by one of the speakers requires an answer from another.

With the appearance of the internet, the way that humans communicate completely changed. A key element for this type of communication is represented by its development in real time, the message becoming an important proportion of spontaneity. Although this brings the internet mediated conversation closer to a traditional one, which takes place face to face, there are some key elements that makes it different from a natural one. Extra verbal means of communication are no longer present in their classical form, being replaced instead by stylistic elements borrowed from classical written communication and by symbolic elements through which different emotions are transmitted. However, online chat conversations allow the existence of more than one thread of discussions in the same time [1] which is a key factor for interanimation.

In the context of the rapid development of computers and increase of computational power, it has become feasible to perform better analysis and processing of texts using computer systems. At the beginning of artificial intelligence, attempts were made to mimic the steps used by a linguist: defining and evaluating the rules of speech and writing by using regular expressions and context-free grammars. With the increase of computational power and of the number of texts in electronic format, it became possible to use powerful statistical techniques in order to study, process, and find meaning in texts. The ambiguity of language found in texts (especially in the colloquial ones) has led to the treatment of natural language processing in a similar manner to random processes, in which the understanding of distributions of probability is essential. Advantages of using this kind of models are the capacity of a system to learn on texts that have not been studied in the past, rising of accuracy by introducing new training data, and the general scalability of the program.

We can use neural networks to build efficient classification models. They are based on the analogy with the structure and function of the human neuron, being systems with the ability to produce results based on statistics without constructing an algorithm or set of rules. Due to the computational power and the huge volume of data, by modeling a structure based on that of the human brain, the purpose of neural networks is to create much better and simpler machine learning algorithms. Neural networks thus offer the opportunity for programmers to replace existing models with some that can perform better, to model through new structures problems of natural language processing and to automatize the process of designing a program through a reduced dependency by a professional linguist. The use of neural networks in the field of natural language processing is currently intensively studied, due to the performance obtained by models that need a large volume of data but do not require linguistic expertise for proper operation [7].

IMPLEMENTATION

The conversations that we analyzed are saved in XML format [4], where element *Corpus* represents the whole content of the file, *Dialog* refers to the discussion in which the participants are engaged, *Participants* relates to the people who communicate, *Person* refers to one of the participants, *Topics* represents the subjects which are discussed, *Turn* cumulates one or more utterances from one individual, and *Utterance* refers to the actual reply in the chat. Besides elements, there are attributes defined, where *genid* is an unique identifier for utterances, *time* refers to the moment when a reply was transmitted, and *ref* describes an explicit link added by a chat participant to a previous reply. This value can be either positive, the value referring to the *genid* of a previous reply, or -1, noting that the author of the current reply did not add an explicit link.

The *convergence* attribute marks the fact that the utterance is in a convergence relation to a certain reply and it can refer to more than one. The *divergence* attribute marks the fact that the sentence is in a divergence relation to a certain reply and, like convergent replies, several previous utterances may be referred to. Below we will illustrate the steps we made for preparing the training and test datasets.

Step 1

First, we parse the whole XML corpus, which consists of 63 annotated conversations and we save convergent or divergent utterance pairs. For this, we identify the replies have the attribute of convergence or divergence defined and the reply for which the link is formed.

Step 2

Then we pre-process the utterances in which expression like "*I'm*" are transformed into "*I am*", "*We're*" into "*We are*". Then all uppercase letters are converted to lowercase, punctuation is removed, and words that are made up of a single letter, irrelevant to the context but also the names of participants are removed. After applying these steps, the words are lemmatized in order to bring each word to its basic form, thus reducing the size of the vocabulary and grouping together words with the same lemma.

Step 3

Because the length of the utterances is variable, from 5-6 words to extremely long ones that have more than 32 words as is showed in Figure 1., their size is limited to a constant value. To further reduce the vocabulary size, words that have a frequency less than a certain value are removed. During testing, this value is 10. It is important to note that very rare words usually do not provide useful information for statistical text classifiers.

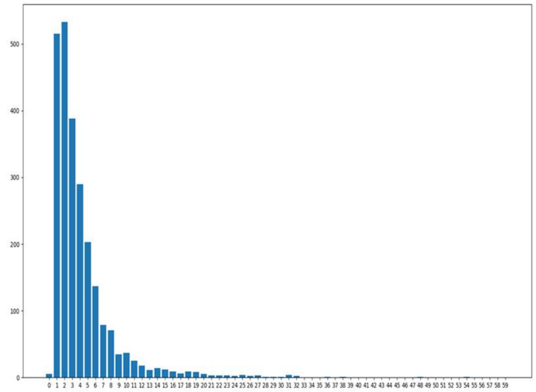


Figure 1. The number of convergent and divergent replies of a certain length

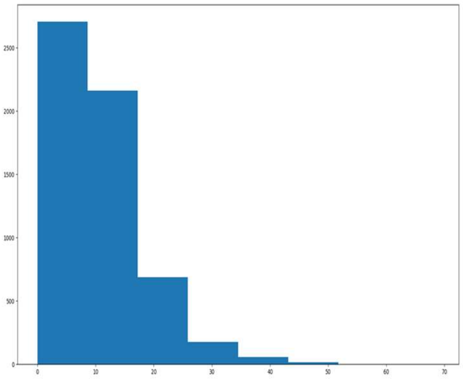


Figure 2. Distances between convergent replies and the utterance for the link is formed

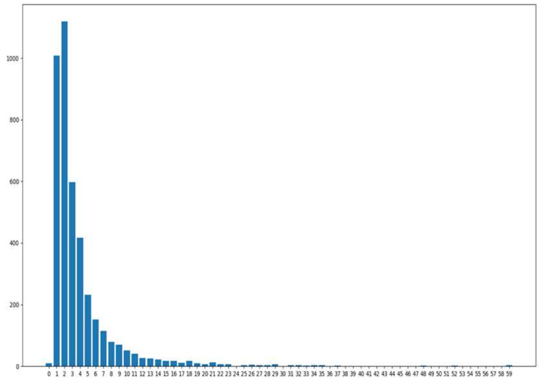


Figure 3. Distances between convergent replies and the utterance for the link is formed

In Figures 1-3 are presented relevant statistics about the utterances in our dataset, including the utterance lengths and distances between replies.

THE 3-CNN MODEL

The method initially chosen to detect convergence and divergence relations in chat conversations is using a 3-channel neural model consisting of Embedding and Convolutional layers (see Figure 4). This model uses only the utterances that have the attribute convergent or divergent defined. Neural networks that use convolutional layers can be used to classify a statement due to their ability to identify words that are important for defining class membership. This Convolutional layer does not take into account the position of the words in the sequence received as input.

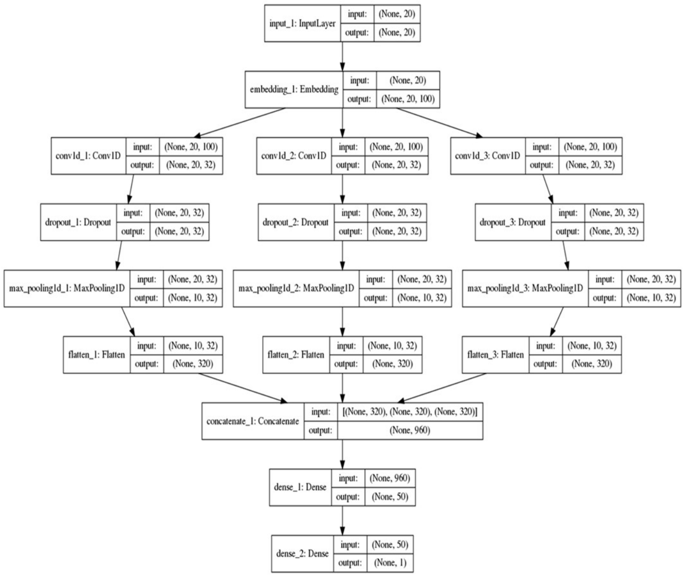

input_1: InputLayer | input: (None, 20) | output: (None, 20)
embedding_1: Embedding | input: (None, 20) | output: (None, 20, 100)
conv1d_1: Conv1D | input: (None, 20, 100) | output: (None, 20, 32)
conv1d_2: Conv1D | input: (None, 20, 100) | output: (None, 20, 32)
conv1d_3: Conv1D | input: (None, 20, 100) | output: (None, 20, 32)
dropout_1: Dropout | input: (None, 20, 32) | output: (None, 20, 32)
dropout_2: Dropout | input: (None, 20, 32) | output: (None, 20, 32)
dropout_3: Dropout | input: (None, 20, 32) | output: (None, 20, 32)
max_pooling1d_1: MaxPooling1D | input: (None, 20, 32) | output: (None, 10, 32)
max_pooling1d_2: MaxPooling1D | input: (None, 20, 32) | output: (None, 10, 32)
max_pooling1d_3: MaxPooling1D | input: (None, 20, 32) | output: (None, 10, 32)
flatten_1: Flatten | input: (None, 10, 32) | output: (None, 320)
flatten_2: Flatten | input: (None, 10, 32) | output: (None, 320)
flatten_3: Flatten | input: (None, 10, 32) | output: (None, 320)
concatenate_1: Concatenate | input: [(None, 320), (None, 320), (None, 320)] | output: (None, 960)
dense_1: Dense | input: (None, 960) | output: (None, 50)
dense_2: Dense | input: (None, 50) | output: (None, 1)

Figure 4. Neural network with 3 CNN channels

Before the convolutional layers, there is an embedding layer with the role of representing a mapping of each word in the vocabulary to a continuous vector space, with similar words being located in nearby regions. Other layers in the network have the role of regularization (Dropout) and sum up the results. We use different sizes for the kernels so we can capture different groups of words, the number of filters is set to 32, and the size of the embedding layer is the same as the length of the utterance. In Figure 4 we provide the complete representation of the architecture.

THE SIAMESE MODEL

The second considered method consists of using a Siamese neural network model that uses Long Short-Term Memory (LSTM) cells [9], inspired by the work of Mueller and Thyagarajan [10]. LSTM cells learn contextual dependences between words in sentences that can pass through the network at a large time difference.

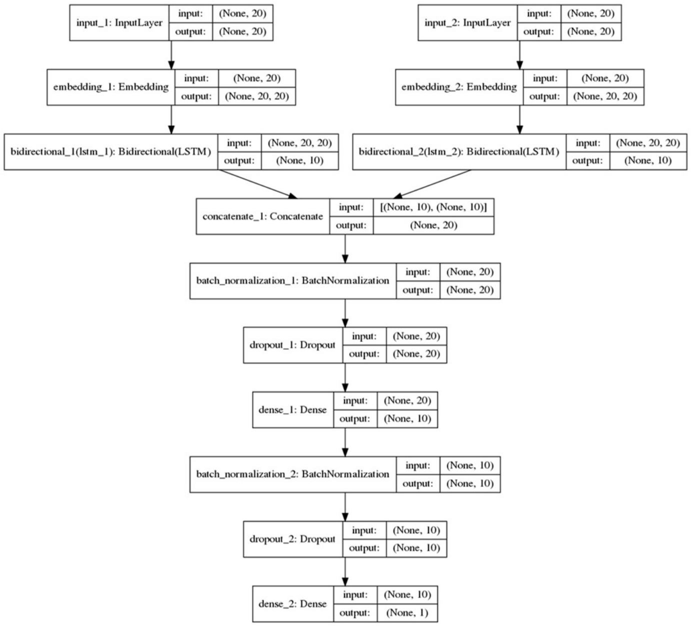

input_1: InputLayer | input: (None, 20) | output: (None, 20)
input_2: InputLayer | input: (None, 20) | output: (None, 20)
embedding_1: Embedding | input: (None, 20) | output: (None, 20, 20)
embedding_2: Embedding | input: (None, 20) | output: (None, 20, 20)
bidirectional_1(lstm_1): Bidirectional(LSTM) | input: (None, 20, 20) | output: (None, 10)
bidirectional_2(lstm_2): Bidirectional(LSTM) | input: (None, 20, 20) | output: (None, 10)
concatenate_1: Concatenate | input: [(None, 10), (None, 10)] | output: (None, 20)
batch_normalization_1: BatchNormalization | input: (None, 20) | output: (None, 20)
dropout_1: Dropout | input: (None, 20) | output: (None, 20)
dense_1: Dense | input: (None, 20) | output: (None, 10)
batch_normalization_2: BatchNormalization | input: (None, 10) | output: (None, 10)
dropout_2: Dropout | input: (None, 10) | output: (None, 10)
dense_2: Dense | input: (None, 10) | output: (None, 1)

Figure 5. The siamese neural network architecture

Our siamese architecture employs bidirectional LSTM cells through which both the original and the inverted sequence are passed, thus defining a context for words from both a past and future perspective. Before the bidirectional LSTM layers, the model uses embedding layers that create a mapping for each word in the vocabulary to a continuous vector space. The model uses the pair of utterances that creates a point of divergence or convergence (current utterance and previous utterance mentioned as a divergence or convergence relation). The size of the embedding layers is the same as the number of the words a reply has and the dimension of the LSTM cells is set to 20. In Figure 5 we provide the complete representation of the siamese architecture.

RESULTS

The corpus of chats used for training and evaluation contains 63 conversations with multiple participants in an educational setting [4]. The dataset is split randomly, using 85% as training data and 15% as test data. The training is done over 100 epochs and the test dataset is used as validation data.

The results using the two neural networks proposed in the previous chapter are shown in Tables 1-4.

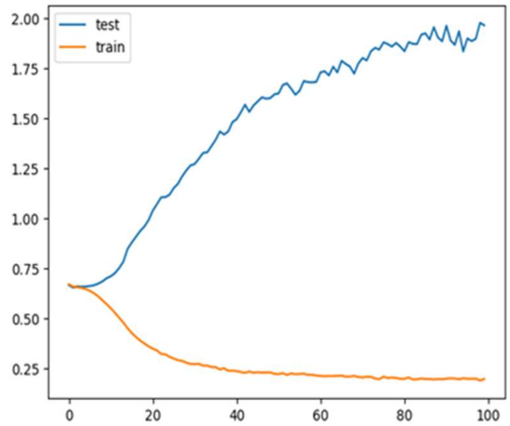


Figure 6. Evolution of the loss function for replies containing 40 words using 3-CNN network.

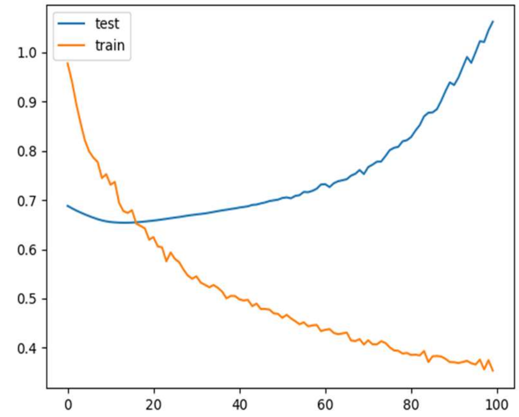


Figure 7. Evolution of the loss function for replies containing 40 words using Siamese network.

From the charts presented in Figures 6-7 that show the evolution of the loss function for both neural networks we can easily see the presence of overfitting. In the case of the Siamese network, the results are slightly better and we can conclude if we use a larger dataset the results the loss on the test data will be lower. Below are the results of loss and accuracy in both models, as well as precision per class. Maximum size of an utterance will be truncated to a dimension of 20, 30, 40 words in order to reduce the size of the vocabulary.

Test/Size Utterance = 20	3-CNN		Siamese	
Accuracy	0.6265		0.645	
Loss	0.6551		0.6487	
Precision	Conv. 0.63	Div. 0.35	Conv. 0.64	Div. 0.37

Table 1 Results for 3-CNN and siamese models using replies that contain a maximum of 20 words

Test/Size Utterance = 30	3-CNN		Siamese	
Accuracy	0.6395		0.6395	
Loss	0.6558		0.6591	
Precision	Conv. 0.62	Div. 0.33	Conv. 0.63	Div. 0.34

Table 2 Results for 3-CNN and siamese models using replies that contain a maximum of 20 words

Test/Size Utterance = 40	3-CNN		Siamese	
Accuracy	0.6395		0.6344	
Loss	0.6589		0.6607	
Precision	Conv. 0.63	Div. 0.35	Conv. 0.63	Div. 0.35

Table 3 Results for 3-CNN and siamese models using replies that contain a maximum of 40 words

Test/Size Utterance = var.	3-CNN		Siamese	
Accuracy	0.6115		0.6395	
Loss	0.6853		0.6541	
Precision	Conv. 0.63	Div. 0.35	Conv. 0.64	Div. 0.36

Table 4 Results for 3-CNN and siamese models using replies that contain a variable number of words

As a comparison, the results using Naive Bayes are shown in Tables 5-10. The entry Freq of the tables denotes that words that appear less than 10 times in the corpus will be deleted or no words will be removed.

Test/NB Monograms	Size Utterance = 20	
Accuracy/Freq < 10	0.66	
Accuracy/ Freq < 0	0.72	
Precision	Conv. Freq<10 0.75 Freq<0 0.76	Div. Freq<10 0.53 Freq<0 0.60

Table 5 Results of Naive Bayes using monograms for replies that contains a maximum number of 20 words

Test/NB Monograms	Size Utterance = 30	
Accuracy/Freq < 10	0.69	
Accuracy/ Freq < 0	0.73	
Precision	Conv. Freq<10 0.78 Freq<0 0.76	Div. Freq<10 0.56 Freq<0 0.64

Table 6 Results of Naive Bayes using monograms for replies that contains a maximum number of 30 words

Test/NB Monograms	Size Utterance = 40	
Accuracy/Freq < 10	0.68	
Accuracy/Freq < 0	0.73	
Precision	Conv. Freq<10 0.78 Freq<0 0.76	Div. Freq<10 0.55 Freq<0 0.66

Table 7 Results of Naive Bayes using monograms for replies that contains a maximum number of 40 words

Test/NB Monograms	Size Utterance = var	
Accuracy/Freq < 10	0.69	
Accuracy/Freq < 0	0.74	
Precision	Conv. Freq<10 0.78 Freq<0 0.76	Div. Freq<10 0.56 Freq<0 0.66

Table 8 Results of Naive Bayes using monograms for replies that contains a variable number of words

Test/NB Bigrams	Size Utterance = var	
Accuracy/Freq < 0	0.76	
Precision	Conv. 0.75	Div. 0.76

Table 9 Results of Naive Bayes using bigrams for replies that contains a variable number of words

Test/NB Trigrams	Size Utterance = var	
Accuracy/ Freq < 0	0.78	
Precision	Conv. 0.78	Div. 0.78

Table 10 Results of Naive Bayes using trigrams for replies that contains a variable number of words

CONCLUSION

The paper proposes two models of neural networks that target the problem of detecting convergent and divergent replies in a collaborative chat with multiple participants. The accuracy of the proposed neural models is low because they tend to overfit. The main cause is the small size of the dataset used for training and the noisy character of the data in chat conversations.

For all these reasons, using Naïve Bayes, a simple machine learning technique, we obtain better results for the task at hand. In the future, by adding additional chats annotated with divergence and convergence relations and thus increasing the data used for training, the Siamese model promises good results.

An important conclusion of this paper is that neural methods are still lacking in performance for small datasets that are often used in education or other social sciences where gathering large datasets is difficult due to the complexity of the task at hand.

REFERENCES

1. Stefan Trausan-Matu (2020) The Polyphonic Model of Collaborative Learning, in Mercer, N., Wegerif, R., & Major, L. (eds.), *The Routledge international handbook of research on dialogic education*, ISBN :978-1-138-33851-7, New York, NY : Routledge, pp. 454-468,
2. Stefan Trausan-Matu, Traian Rebedea (2010), A Polyphonic Model and System for Inter-animation Analysis in Chat Conversations with Multiple Participants, in A. Gelbukh (Ed.), *CICLing 2010*, LNCS 6008, Springer, 2010, pp. 354–363
3. Puccio, G. J., (1998). *Letters from the field*. Roeper Review, 21, 85-86.
4. Stefan Trausan-Matu, Mihai Dascalu, Traian Rebedea, Alexandru Gartner, Corpus de conversatii multi-participant si editor pentru adnotarea lui, *Revista Română de Interactiune Om-Calculator*, Vol.3, No.1, 2010, pp. 53-64, MatrixRom
5. Runco, M. A., (1990). *The divergent thinking of young children: Implications of the research*. Gifted Child Today, 13, 37-39.
6. Runco, M. A., and Albert, B. S., (1989). *Independence and the creative potential of gifted and exceptionally gifted boys*. Journal of Youth and Adolescence, 18, 221-223.
7. Brownlee J., (2017). *Deep Learning with Python: Develop Deep Learning Models on Theano and*

TensorFlow Using Keras. Melbourne, Australia: Machine Learning Mastery.

8. Mohammad Hamad Allaymoun (2016). *Analysis of rhetorical, altruistic convergent and divergent dimensions in CSCL chats*, Ph.D. Thesis.
9. Hochreiter S., Schmidhuber J. (1997). *Long short-term memory*. Neural computation. 9(8):1735-80.
10. Mueller J., Thyagarajan A., (2016). Siamese recurrent architectures for learning sentence similarity, in *Thirtieth AAAI Conference on Artificial Intelligence*.

Controlling a programming environment through a voice based virtual assistant

Sonia Grigor
Technical University of Cluj-Napoca, Computer-Science Department
Cluj-Napoca, Romania
sonia.grigor@student.utcluj.ro

Constantin Nandra
Technical University of Cluj-Napoca, Computer-Science Department
Cluj-Napoca, Romania
constantin.nandra@cs.utcluj.ro

Dorian Gorgan
Technical University of Cluj-Napoca, Computer-Science Department
Cluj-Napoca, Romania
dorian.gorgan@cs.utcluj.ro

DOI: 10.37789/rochi.2020.1.1.18

ABSTRACT

The use of smart personal assistants is intended to provide a solution that employs the voice interaction model in order to help improve accessibility for everyday tasks. This model is more than suitable for simple tasks, such as internet searches or device-controlling commands. In this paper we explore the possibility of using this interaction model for completing more complex, composite, context-dependent tasks. Particularly, we look into the potential benefits of using custom spoken commands to help novice users develop insight into the workings of a computer program. Throughout this paper, we present a solution based on an existing, customizable voice assistant that is meant to both help users grasp the structure of a program and improve accessibility for programming activities. The latter is achieved by providing a framework for a voice-based programming environment, offering features like code fragment insertion, navigation, error detection, handling and program running, while also providing voice and text-based feedback for the executed commands.

Author Keywords

Voice-based interaction; Programming assistant; Amazon Web Services; Echo Dot; Alexa;

ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces. H.3.2. Information Storage and Retrieval: Information Storage.

General Terms

Human Factors; Design.

INTRODUCTION

During the last decades, solutions developed by the IT (Information Technology) industry have become ever more prevalent within every kind of human activity, be it personal, social or economic. Computer programming is at the core of this industry and, because of this, software developers are required in order to create the applications impacting our lives. However, computer programming is one of the more challenging occupations, since it normally requires multiple levels of specialized education to adequately master.

The ability to program, in some form or another, is one of the most sought-after skills at workplaces around the world. This high demand for software developers can incentivize many individuals to follow such a carrier path. Nevertheless, acquiring and developing programming skills can be difficult for users with experience in areas other than software development, who lack the basics of a formal education and training. Possible ways to alleviate this problem might include the training of candidates through exposure to practical use-cases and basic solution implementations. This should be done in an intuitive manner, using simplified terminology and real-world, applied examples, while minimizing the use of high-level, abstract concepts.

Recent development in human-computer interaction have led to the rise of intelligent personal assistants. Being controlled through natural language, they offer a human-centered interaction model that is intuitive and easy to use, with little to no training required on the user's part.

The project presented within this paper intends to capitalize on the intuitive nature of the language based interaction model in order to facilitate the training of novice programmers. The idea is to employ an intelligent voice assistant and integrate it with a programming environment, thus allowing for voice commands to be used in controlling code insertion, editing and execution. The proposed solution would employ an intuitive, top-down approach, starting from the general structure of a program, and gradually working towards more specialized constructs and instructions.

The system provides a small and well-defined set of voice commands, with which the user would be able to learn to employ the most important syntax elements of a given programming language. The interaction between the user and the system would be performed on several channels. The system receives input from the user in the form of a spoken command, and then it provides two types of feedback: voice feedback through the response that notifies the user of the status of the executed command, and visual/text based feedback consisting of state changes within the programming environment and the supplying of information or error messages.

RELATED WORK

Human-computer interaction has been studied since the advent of user interfaces and is closely related to the term of usability. This specifies the degree of satisfaction of a user with regard to his interaction with an application through a dedicated interface. In the past, the term computer-human interaction, placing emphasis on the computer, was used to describe the relationship between the two. Nowadays, with the evolution of technology and the attempt to reduce the time and effort required to effectively use a system, the emphasis is on the human element, with the term human-computer interaction accurately describing the current trend [1]. This is most obvious with the recent advent of handheld devices featuring personal assistant software. These are meant to facilitate the interaction of the human with the computer, exploiting increasingly powerful devices to mimic the natural human communication capabilities. Relevant examples include complex processing tasks, such as image processing and classification, text-to-speech and voice recognition.

Virtual assistants

Voice assistants are software agents developed to intercept the human voice, interpret it and respond in a synchronous manner. Each of the biggest players on the IT market have developed one or more intelligent personal assistants to be delivered in different forms and with different roles to fulfill:

- embedded in the phone: *Siri*, *Google Assistant*;
- operating system functions: *Cortana*;
- dedicated devices:
  - *Alexa* embedded in smart speaker Echo devices
  - *Google Assistant* as part of *Google Home* home-automation devices [2].

Alexa

Today, the concept of an intelligent personal assistant is often associated with a smart speaker that can be activated by voice interaction using a wake word, and can perform a variety of user queries [3]. This is the case of *Alexa*, the virtual assistant developed by *Amazon.* It responds to the activation word "Alexa", and offers a sizeable set of standard commands, as well as the possibility for user customization. *Alexa* is the name of the voice service that powers *Amazon Echo* (the intelligent speaker), offering features or abilities that allow customers to interact with devices in a more intuitive, voice-based manner. The provided functionality includes, but is not limited to: internet searches, media playback and device control commands.

Figure 1 shows the method of user interaction with *Amazon Alexa*. First, users produce an utterance or request, which is filtered by *Alexa* through speech recognition, machine learning, and natural language processing. All of these are complex processes, requiring significant computational power and are therefore performed in the Cloud. Alexa then accesses web-hosted services, which employ this functionality, and provides a response to the user. Included in the response process, Alexa produces an information "Card" that records users' words and the resulting system response. The "Card" information is available to users via the *Alexa* application in a textual form, providing a record or interaction history [4].

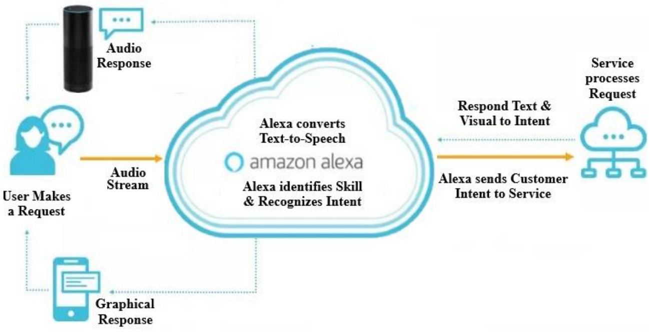

Audio Response
Service processes Request
Respond Text & Visual to Intent
Alexa converts Text-to-Speech
amazon alexa
Alexa identifies Skill & Recognizes Intent
User Makes a Request
Audio Stream
Alexa sends Customer Intent to Service
Graphical Response

Figure 1. Alexa interaction model [4]

Google Assistant

The virtual assistant provided by *Google* is available on smartphones and smart home devices. Its activation words are "Hey Google", or "OK Google" [5]. When the software is activated by voice, the user usually receives feedback in the form of a changed screen if it is a phone, or in the form of a light on the device, if it is a *Google Home* device. On mobile devices it is employed for device control purposes and user queries. Hands free phone operation and internet searches are among the most well-known use cases. In the case of *Google Home*, the assistant is most often employed for device control purposes, managing various types of home appliances [6].

Siri

The *Siri* virtual assistant developed by *Apple* is available on all *Apple* devices including *iPhones*, *MacBooks*, *iPads* and *Apple Watches* [7]. It responds to the activation word "Hey Siri" and uses voice queries and a natural language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of Internet services. Its voice recognition engine was provided by *Nuance Communications*, and uses advanced machine learning technologies to operate [8].

The ways in which *Siri* can be used are diverse and numerous. From common features such as initiating a call, sending a message, to unit conversions, and displaying notifications based on a user device's location, *Siri* finds applications in many areas.

Some specialized commands that *Siri* knows to execute include: adding a post on *Twitter* or *Facebook*, searching for a person's tweets, solving mathematical operations and setting alarms based on a given location, which is activated when the respective location is reached, the conversion of the units of measurement [9].

Cortana

Cortana is the personal assistant developed by *Microsoft* that helps users save time and focus on relevant aspects of utilizing an operating system. It is notable in its ability to gather data on user activity and preferences in order to improve user interaction. Thus, coupled with regular updates to the operating system being pushed by *Microsoft*, give it the potential to continually improve. Some of *Cortana*'s functions include: calendar management, joining a meeting using *Microsoft Teams*, setting alarms, opening applications on the user's computer, help with system's management and settings and more [10].

Similar applications

Interactive programming learning methods are either text-based or in the form of video tutorials available online. They provide potential students with the theoretical information and the opportunity to apply said information.

The authors of [11] developed an application to help beginners learn SQL. They developed both a graphical interface of the application, and provided a natural language processing engine. The paper presents the *Cyrus* application which has two main modes. Firstly, it acts as a guide, allowing students to choose a database on which to experiment with different queries, which can be specified vocally. Secondly, in its knowledge assessment mode, the system allows its users to filter questions based on difficulty levels and answer them through a text-based interaction model. While employed in tutorial mode, the system accepts the voice query in English, maps the query to SQL and executes it to produce the result. As a matter of redundancy, students can also write or edit the SQL themselves, bypassing the voice command interface.

Voice Coder [12] is an extension to *Alexa's* rule set that helps users create games using through voice commands. The game begins with no rules or logic. The user's main responsibility is to program rules, using events, activities, and values. Example of activities include moving or playing a sound.

Coder [13] is another example of an *Alexa* extension. It aims to teach its users programming by providing coding examples. It has support for more than 10 languages (some of them still in progress). Users can ask for examples and instructions on how to write code in some of the most popular languages.

C Programming Quiz [14] is a quiz based on questions about the C programming language. It has instructions for navigating between questions and it was created using available templates provided by *Alexa Skill Kit*.

CS Guru [15] provides users with a selection of questions from the Data Structures and Algorithms field, and has a weekly updated content. Question are straight-forward and they have answers and explanations that can be easily understood and reproduced. This application provides a training mode which is designed to teach users about the key concepts in Computer Science.

SYSTEM OVERVIEW

The project presented in this paper aims to demonstrate how voice interaction with intelligent personal assistants can improve a tool's accessibility by simplifying the way in which certain tasks can be performed. In this context, the term "intelligent assistants" refers to specialized software using a knowledge base to process voice commands by employing sophisticated methods and algorithms for speech recognition and interpretation.

The project described within this paper explores the possibility of using voice interactions to help users learn the basics of programming through natural language. It also aims to provide the basis for a programming environment featuring improved accessibility, which might cater to users suffering from various motor disabilities. The system supports four different programming languages at the moment, with mechanisms in place to allow for easy extension. It permits any user, without knowledge of the system's workings to upload language elements structured in a specific manner, as well as predefined error messages for various situations which might appear when compiling the code.

The objectives of the application are linked to the needs of novice users in the field of programming. The application can meet several needs of such a user, related to learning about: the structure of a program, control structures (conditioned, repetitive or iterative), sorting algorithms like BubbleSort, MergeSort, and the logic of a program.

The overall architecture of the system is shown in Figure 2.

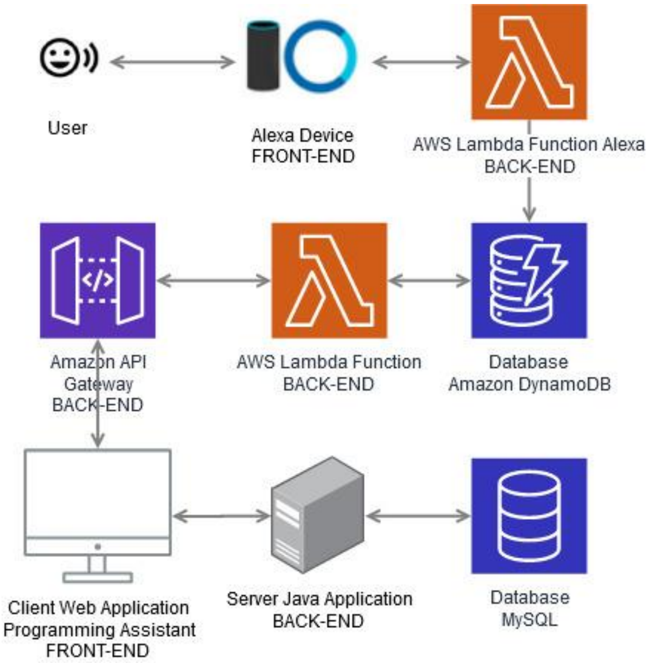


Figure 2. System conceptual architecture

The user interacts with the device hosting Alexa. This is called Amazon Echo, and represents a line of smart speakers sold by Amazon. They can be controlled by voice and include the *Alexa* virtual assistant. The custom command interpretation logic of the assistant resides within a repository of *AWS Lambda Functions*, hosted remotely within *Amazon*'s Cloud. This is accessed by *Alexa* via exposed web services every time that it needs to process a custom command. This logic is pretty low level, is implemented by the user and (in our case) consists in generating a specified code and a number of parameters for each command. These generate data are then stored within the *DynamoDB* database, also hosted by *Amazon*. This database essentially creates a log of all the commands (coded in a specific manner by the user) launched within a given interactive session. By accessing this log, a user application can then programmatically react to each command.

Java Server Application

The server application is meant to control the access to a repository of files containing relevant code fragments and error information for the supported programming languages. These files contain a selection of common programming constructs for each supported language, as well as sets of hints to help users solve specific errors encountered during the compilation process. Access to the code fragments and error information data is provided through a set of web services, making it possible for experienced users or system administrators to modify existing data and even add support for new languages after the system's deployment.

Client Web Application

The client is a web application that provides the user with the visual feedback for the actions resulting from the spoken commands. This is essentially a prototype programming environment that integrates the voice command processing capabilities of the *Alexa* assistant. This results in most of its functionality being directly controllable through voice commands.

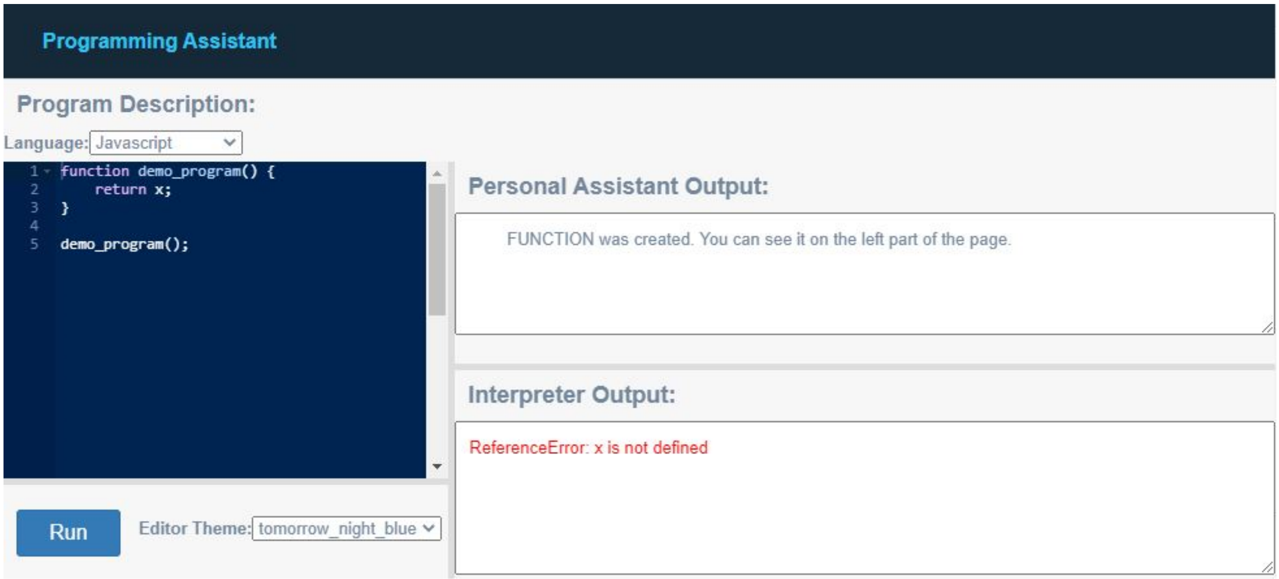
Voice control of the application is achieved by having it retrieve the commands registered within the *DynamoDB* database and applying a series of processing and interpreting steps. These read the name and parameters of the commands logged within the *DynamoDB* database and generate user-specified behaviors. In our case, these behaviors are the actions required to control the programming environment.

Accessing the data logged within the *DynamoDB* database can only be achieved through the *Amazon API Gateway* and *AWS Lambda Function Backend*, which are the mechanisms put in place by Amazon to enforce access control and security.

The graphical user interface of the client application consists of three major areas which can be seen in Figure 3. The first one (left side) contains a code editor where the code is inserted after the relevant commands are processed. A user can select a programming language from the top of the page, and also select a theme from one of the editor themes, located at the bottom of the area. The second area (top-right) provides a log that lists the virtual assistant's feedback. The feedback is chronologically presented so the user can see the order of the added instructions and their results. In the last area (bottom-right) there is the output of the interpreter/compiler. This area contains the result that can either be an error or the expected output of the executed code.

Grammar of interaction model

The grammar of the commands used consists of nine main elements. These can be grouped into two categories: code management and application management. Those from the first category are related to creating programs, variables or operators, navigating through the code, inserting code snippet such as "if", "while", "bubble sort", and printing text. Those from the second category are related to: application start, helping with errors, running programs, wait actions and setting the feedback length/complexity.



```
function demo_program() {
    return x;
}

demo_program();
```

FUNCTION was created. You can see it on the left part of the page.

ReferenceError: x is not defined

Figure 3. User interface of the integrated programming environment

The *start* command, is used to activate the application. The instruction responsible for managing the errors that appear after running the program is *help error*. The *run* command is used in order to start running the code and obtain a result from de compiler. *Wait*, used for increasing the time period in which *Alexa* is active, can be said when the user needs more time and doesn't want the skill to stop. The *set_reponse* command is used when the user wants to select the type (length) of voice feedback to be received from *Alexa*. The latter can be long – *Alexa* details the action, short – *Alexa* says only that it intercepts the command, and no – *Alexa* doesn't give any feedback.

The *create* command is used to add a new program template which has a specific name. Also, with this one, the user can add a new named variable or an arithmetic operator like *plus*, *minus*, *multiply* or *divide*. *Go to* permits navigating to different lines of code or getting into different named functions. *Insert* lets the user add templates for different programming statements like "if", "switch", "for" and "while". Also, using this command, users can add sorting algorithms. In order to print a sentence on the screen, the user need to say *write* followed by what he wants to print.

The grammar is a simple one but offers the most common commands in terms of a programming language. Simplicity is an essential feature of the dialogue model in such a situation, for a novice user. The user can express himself briefly and concisely through a well-defined set of keywords. The grammar has several levels of expansion. The rules consist of a terminal (uppercase) and a non-terminal (lowercase). The terminals have synonyms, so the user has multiple way to interact with the system without being constraint to use a strictly set of words.

```
command ::= start | create | go_to | insert | HELP ERROR
            | write | RUN | WAIT | set_response
create ::=CREATE OPERATOR operator_name
            |CREATE  PROGRAM program_name
             |CREATE  VARIABLE variable_name
go_to ::= GO TO LINE number
            | GO TO FUNCTION program_name
insert ::= INSERT structure | INSERT FOR number
structure ::= IF | SWITCH | WHILE | DO WHILE
            | BUBBLE SORT | MERGE SORT
set_reponse ::= SET RESPONSE response
response ::= NO | SHORT | LONG
```

SYSTEM EVALUATION

This section presents an evaluation of the system's functionality and performance. The tests were designed to verify functional requirements of the system, while also validating the interaction between the system's components.

Performance Evaluation

The performance evaluation of the system was done by analyzing its average response time for user requests. In this context, a user request refers to the *HTTP* request submitted from the user interface (through Alexa and the associated command interpretation logic) to the two application servers: the command history server, deployed on *Amazon's* infrastructure, and the server hosting the language construct and error data, hosted on the local machine. To evaluate their performance, a few endpoints were chosen to measure the time required for a response to be received. To get accurate results and level out any irregularities, each server was subjected to five calls for the same endpoint, and the response times were averaged. The results obtained can be seen in Figure 4.

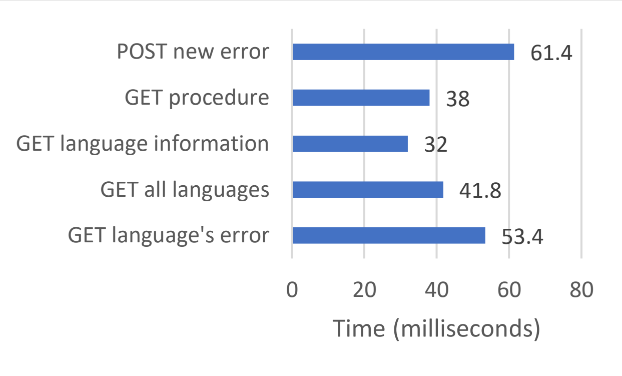


Figure 4. Response time of local server

Referring to Figure 4, which presents the performance evaluation of the local server responsible for delivering language syntax and error data, the first test was performed to add a new error interpretation message and the call requires the longest amount of time to complete (61.4 ms). This result is to be expected, as the request relies on a POST method that contains an object with 4 parameters in the body of the message, so it requires more processing time. The following methods are of the GET type. The second call was performed in order to measure the time required to extract detailed information on specific errors. The average response time was 38 ms. Analogous to the second test, the third was created to request the data in the form of syntactic constructs for a given programming language, resulting in a similar processing time of 32 ms. The call to fetch the syntax constructs data from all the registered languages within the database takes an average of 41.8 ms, and is reasonable because the result is in the form of a list of four elements of a complex data type. The last test, aimed at measuring the time required to fetch the sets of errors associated with a language, averages a response time of 53.4 ms.

Figure 5 shows the average response times for the *NodeJS* server developed using *AWS Lambda* and hosted on *Amazon's* infrastructure. The first test was performed to verify the time required to insert an error into the NoSQL *ErrorLearnProgramming* database, with a resulting average time of 133.6 ms. The second test consists of deleting the entire database that contains the user's history of programming errors. As a sequential deletion was required, a longer response time (286 ms) was expected. The last test shows the performance of the server in terms of reading databases. This reading is an action that is performed very often, since the system polls the database to monitor changes in the command history. The average access time is 266.2 ms which is a reasonable time considering that oftentimes, the response consists of an array of approximately 30 complex object elements.

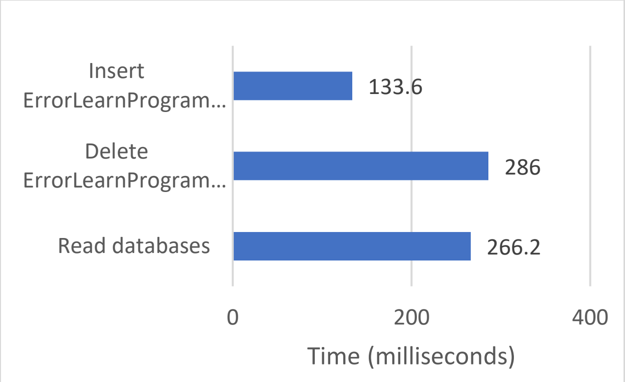


Figure 5. Response time of AWS Lambda

Usability Evaluation

The ability of the system to behave as expected, to avoid unwanted command errors or to lose control is assessed below. In this analysis, the fulfillment of some use case scenarios was monitored, in an increasing order of complexity.

The tests included a variable number of instructions to analyze the dependence between the complexity of the scenario and the number of errors occurring. By error, in this context, we mean a situation in which *Alexa* did not understand the command and/or failed to act appropriately. Three types of scenarios were tested: low complexity (5 commands), medium complexity (15 commands) and high complexity (25 commands). These scenarios do not necessarily involve completely distinct command, but lead to distinct outcomes.

Figure 6 shows the influence of the number of commands on the number of errors. When referring to errors, it must be understood that they are errors of control of the system or errors of misunderstanding of the spoken words. In the case of the scenario containing 15 instructions, only one error occurred. This was related to semantics, more precisely the sentence was not understood by the vocal assistant. As for the last scenario consisting of 25 commands, the two errors occurred, also at the semantic level.

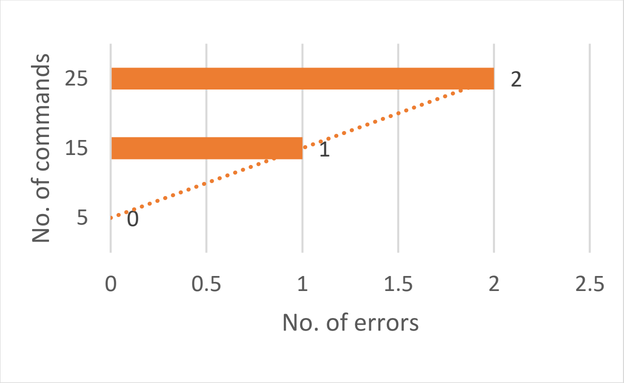


Figure 6. Commands - errors dependency

Figure 7 shows the relationship between the number of commands and the time required to fulfill them. As can be observed, the total time required for the completion of a scenario increases linearly, while the average time for realizing a command marginally fluctuates around the 10 seconds value. These measured time intervals include both the time required for the utterance of the command and that necessary for *Alexa's* spoken feedback. It should be noted, however, that we instructed *Alexa*, at the beginning of the test, to provide only the strictly necessary information. Therefore, the feedback messages were as short as possible.

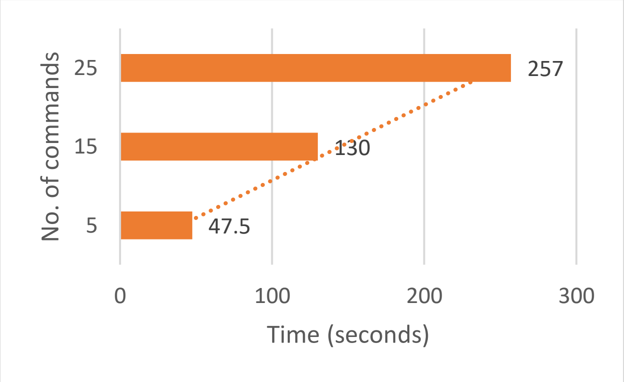


Figure 7. Commands - time dependency

Heuristic Evaluation

Heuristic evaluation has been defined as a usability engineering method that aims to identify usability issues in the design of a user interface.

An initial set of nine heuristics is given by Molich and Nielsen in 1990. After that, Nielsen in 1994, refined these heuristics and proposed a set of 10 usability principles [16]. This section contains an analysis of the system from the point of view of Nielsen's ten heuristics.

Visibility of system status - Information about the system's status and command execution is provided to the user in a reasonable amount of time. The mean observed time for the execution of a command is 10 seconds, from the utterance of the command to the voice feedback received from Alexa. The system can provide redundant, text-based feedback, containing additional information. Both types of feedback are supplied at the same time – immediately after the execution of the command. There is also the visual feedback, observed through the changes affected by the commands regarding cursor movement, code insertion, code folding and so on.

Match between system and the real world - The system uses terms that a novice user in the field knows, uses simple words and familiar concepts providing explanations where needed. This heuristic is applied in terms of verbal dialogue between the user and *Amazon Echo Dot*, but also in the web application by providing information in an easy to understand form.

User control and freedom – This is accomplished through several means: the system gives the user the opportunity to choose how he wants to be given the feedback, offers the possibility to do undo and redo commands and can help the user navigate through and solve errors. Another feature is the ability to extend the time Alexa listens to the user by using the wait command, or stop the interaction entirely by employing a built-in command.

Consistency and standards – The consistency of the system is respected both at the level of voice commands by employing the same keyword to address similar operations, minimizing the size of the keyword vocabulary and at the textual feedback level, following a standardized notation and structure for both the information and error messages.

Error prevention - In this regard, the system does not provide much support, being susceptible to command interpretation errors. However, it provides visual feedback when writing a code sequence that does not follow the language syntax and grammar rules. This is displayed within the programming environment.

Recognition rather than recall - This heuristic is quite difficult to apply within a voice-based interaction model. However, there is little support from the system, reminding the user to select the type/length of feedback to be delivered, as well as select the desired programming language to work with. These prompts are delivered by the system at the start of a working session, thus relieving the user from the need to remember the format of these particular commands.

Flexibility and efficiency of use - The accelerators made available by the system that make an expert user more efficient are the following: the option of not having a voice response (thus minimizing overall interaction time) and the possibility of uttering several commands at a time joined together by the particle *and*. As of this moment, up to two joined commands have been consistently observed to be executed correctly by the system.

Aesthetic and minimalist design - The dialogue required to achieve the desired effects is minimal without the need to provide non-essential information. The system does not ask the user for irrelevant or unnecessary information. Also, the visual interface of the programming environment has been kept as simple and clean as possible – as seen in Figure 3.

Help users recognize, diagnose, and recover from errors - System error messages are presented in textual form, without using encodings. Furthermore, the system is capable of providing the users with hints associated to each generated error. These hints are user-customizable and can thus be modified, without altering the system's implementation.

Help and documentation – At the moment, extensive documentation for the system is not available, as it is an ongoing work. However, provisions can be made to integrate a documentation within the webpage hosting the development environment and future developments can see the use of *Alexa* to navigate and find items of interest based on voice commands.

CONCLUSION

The system described within this paper aims to take advantage of the intuitive and easy to use voice-based interaction model in order to facilitate the teaching of basic programming concept to novice users. To this end, it employs Amazon Alexa's API for creating custom voice commands (also referred to as skills). The proposed command set is limited in size and emphasizes language simplicity and compactness.

The contributions of the implemented project include the integration of an existing intelligent voice controlled assistant within a programming environment, the design and implementation of a set of simple and concise commands to manage the interaction between the user and the system and the development of a scalable support infrastructure allowing for the addition of multiple programming languages.

Thorough this paper, the basic architecture, components and functionality of the system are presented along with an evaluation of its performance – in terms of server response time – and usability. The latter is expressed in terms of the system's average command execution time, errors per command ratio and also by considering Nielsen's ten usability heuristics.

REFFERENCES

1. Kumar, R., "Human Computer Interaction", Laxmi Publications, (2008), ISBN: 978-8131802809
2. Hoy, M., "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants," Medical Reference

Services Quarterly, vol. 37, pp. 81-88, (2018), doi: 10.1080/02763869.2018.1404391

3. Miluț, C., Iftene, A. and Gîfu, D., "Iasi City Explorer - Alexa, what can we do today?" in Proceedings of RoCHI - International Conference on Human-Computer Interaction, pp. 139-164, (2019)**,** ISSN 2501-9422
4. Lopatovska, I., "Overview of the Intelligent Personal Assistants", Ukrainian Journal on Library and Information Science, pp. 72-79, (2019), doi: 10.31866/2616-7654.3.2019.169669
5. Google, "Google Assistant, your own personal Google", (2020), https://assistant.google.com/
6. Hadi, M. S., Shidiqi, A. A. and Zaeni, I. A. E., "Voice-Based Monitoring and Control System of Electronic Appliance Using Dialog Flow API Via Google Assistant," 2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Denpasar, Bali, Indonesia, pp. 106-110, (2019), doi: 10.1109/ICEEIE47180.2019.8981415.
7. Matei, A., and Iftene, A., "Smart Home Automation through Voice Interaction," in Proceedings of RoCHI - International Conference on Human-Computer Interaction, pp.132-137, (2019), ISSN 2501-9422
8. Apple, "Siri - Apple,", (2020), https://www.apple.com/siri/
9. Aron, J., "How innovative is Apple's new voice assistant, Siri?", New Scientist - NEW SCI, vol. 212, pp 24-24, (2011), 10.1016/S0262-4079(11)62647-X
10. Microsoft, "What is Cortana?", (2020), https://support.microsoft.com/ro-ro/help/17214/cortana-what-is
11. Godinez, J. E. and Jamil, H., "Meet Cyrus: The Query by Voice Mobile Assistant for the Tutoring and Formative Assessment of SQL Learners", SAC '19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 2461-2468, (2019), ISBN: 978-1-4503-5933-7, doi: 10.1145/3297280.3297523
12. Dickinson, J., "Amazon.com: Voice Coder: Alexa Skill", (2020), https://www.amazon.com/Jimmy-Dickinson-Voice-Coder/dp/B07HFWQPKN
13. Fireberger, A., "Amazon.com: Code: Alexa Skill", (2020), https://www.amazon.com/aviram-fireberger-Coder/dp/B07P81FZVL
14. Manish, A., "Amazon.com: C Programming Quiz: Alexa Skill", (2020), https://www.amazon.com/Manish-A-C-Programming-Quiz/dp/B07Z514B3F
15. Dephony, "Amazon.com :CS Guru: Alexa Skill", (2020), https://www.amazon.com/DEPHONY-CS-Guru/dp/B07VRF5BKR
16. Pribeanu, C., "Tendinte actuale în evaluarea interfetelor om-calculator," Informatica Economica, vol. 2 nr. 4(8), pp. 21-25, (1998), ISSN 1453-1305

Increasing Diversity with Deep Reinforcement Learning for Chatbots

Cristian Pavel
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
cristian.pavel@stud.acs.upb.ro

Ştefania Budulan
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
stefania.budulan@cs.pub.ro

Traian Rebedea
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
traian.rebedea@cs.pub.ro

DOI: 10.37789/rochi.2020.1.1.19

ABSTRACT

Dialogue generation for open-domain conversations is a difficult and open problem that, so far, has not been able to approach human-level performance. Recently, a popular solution is to apply a sequence-to-sequence architecture, similar to the machine translation problem. These models try to map the input - given as the previous utterances, to the output - the next utterance. Unfortunately, they usually tend to repeat sentences, often preferring dull responses, that end the conversation abruptly. Therefore, Reinforcement Learning techniques have been combined with the standard sequence-to-sequence models in order to avoid their shortcomings. Our model applies a Policy Gradient method that maximizes the expected reward of generating the next utterance given a history of previous utterances. The results show an improvement in diversity up to 0.16 - almost 10x higher than the model without RL, while keeping the responses relevant to the input message.

Author Keywords

Dialogue generation; Reinforcement learning;
Conversational agents; Sequence-to-sequence model.

INTRODUCTION

A simplistic approach in dialogue generation for conversational agents uses a supervised learning method that tries to generate the next turn of a conversation, given a subset of the previous turns. This approach fosters some downsides. Firstly, it uses a Cross-Entropy Loss, which suffers from exposure bias [3] and has no quantifiable relation with the traits that healthy dialogues should have (e.g., informativeness, engagement, or diversity). Reinforcement Learning (RL) manages to overcome these problems by using rewards, aiming to guide the model towards an action space that is consistent with generating a human-like dialogue.

The main focus of this work is the problem of low diversity, representing the tendency of a model trained with Cross-Entropy Loss to output generic responses such as '*I don't know*' or *'I have no idea*' [12, 19]. This issue appears mainly from the sparsity of the data and the high number of inputs that a generic response can match [18].

Our proposed solution resides on the work of Li et al. [13]. We use three models, one (the Reinforcement Learning model) leveraging the other two (sequence-to-sequence) to compute rewards and update its actions. The standard sequence-to-sequence architecture [4, 23] stands as a building block for all the models. The REINFORCE [26] algorithm is used in the training iterations of the final network, which is initialized from a pre-trained chatbot in a supervised manner.

In the context of Human-Computer Interaction, chatbots can pave the way to artificial general intelligence. A survey of the vast number of conversational agents developed in recent years has been performed by Grudin and Jacques [10]. As the survey authors emphasized, constructing an open-domain chatbot is an arduous task and the present work is another proof of that statement. In addition to this, Allen et al. [] analyze the possibility of using a practical dialogue between the user and the system as the main mechanism connecting the two and hypothesize that such a user interface can replace the current popular Graphical User Interfaces (GUI) in the future. Moreover, Følstad and Brandtzaeg [] discuss in more detail the implications, challenges and opportunities that emerge when transitioning towards natural language interfaces and chatbots, a next step which is predicted by multiple tech companies, according to the authors.

RELATED WORK

Neural response generation has been extensively studied in the last few years, starting from the novel idea of Ritter et al. [17] who propose the application of Statistical Machine Translation (SMT) techniques to the problem at hand.

Following this approach, due to the success of the sequence-to-sequence (seq2seq) architecture [4, 23] for the Neural Machine Translation problem, this method has been rapidly transferred to the dialogue generation task [17, 25]. These simple networks manage to respond coherently and even preserve some context, without any prior knowledge or pre-engineered rules. These results motivate our choice of the seq2seq architecture with a Recurrent Neural Network (RNN) based encoder and decoder. In a similar direction, an end-to-end approach is tackled by Sordoni et al. [22]. They propose three different methods to incorporate the context of the conversation into the generation procedure and they decide to use their model as an extra feature to the SMT systems reaching an improvement over the considered baselines. We also experiment with their intuition of concatenating the context to the current message and then pass the result to the encoder.

In overcoming the lack of diversity of these models [12, 19], there have been proposed several solutions. A variational auto-encoder can be used to add additional variance into the model [20]. Similarly, promising results have been obtained by using a Generative Adversarial Network (GAN) usually combined with Reinforcement Learning (RL) to backpropagate the error from the discriminator to the generator [14]. Moreover, Li et al. [13] utilize Reinforcement Learning with heuristic rewards that try to capture relevant attributes of a dialogue and increase considerably the diversity of the baseline. The model in our paper also makes use of these rewards and their architecture stands as a starting point for our implementation.

The Transformer architecture [24] has shown great potential in Natural Language Processing (NLP), especially with the emergence of BERT [8] and the possibility to fine-tune this architecture depending on the task at hand. Recently, this robust model has also been applied to the dialogue generation task. One example of this is the Meena chatbot [1] that outperforms previous well-known chatbots such as Cleverbot [5] or Xiaoice [27]. The authors also propose a new human evaluation metric, Sensibleness and Specificity Average (SSA), that incorporates both sensibility and specificity and show that this metric is correlated with perplexity. For the current experiments, we do not utilize a Transformer network, but future work can aim to improve our results by incorporating a Transformer-based seq2seq model.

DATASET

The dataset we used for training is called Cornell Movie Dialogs [7] and it contains metadata-rich exchanges extracted from various movies. There are 221,282 sentence-response pairs accompanied by information about the speakers involved and the movies in which each exchange takes place. Figure 1 shows the distribution of the utterances' lengths in the dataset. The choice of this dataset is motivated by its relatively small size compared with other datasets (e.g., OpenSubtitles [15]) while also being easy to parse and use. It also contains less noise and thus a model can be trained without needing too many input-output pairs. Training on such a small dataset we do not set about or expect to construct a state of the art final model. The goal remains to tackle the diversity problem, while being able to respond to simple input messages. We chose to eliminate the input-output pairs in which either the message or the response had more than 10 tokens. After this elimination we remain with approximately 28% of the data.

Because the concerning issue is related to generic utterances, Table 1 shows the most frequent sentences in responses from the training data, multiplied by the number of different bigrams in the input messages and scaled by the total number of bigrams in the vocabulary. This is done to differentiate frequent input-output pairs from generic responses that fit multiple different inputs. These responses will later be used in the calculation of the rewards.

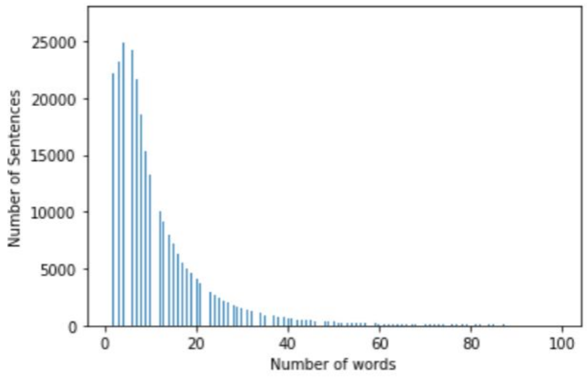


Figure 1. Histogram showing the distribution of the lengths of the utterances in the dataset

Table 1. Frequent sentences in responses from the training data

Response	Scaled Frequency (x 10^{-10})
I don't know.	1.32
Yeah.	1.28
Well.	1.17
I.	1.12
No.	0.76
Yes.	0.64
What?	0.61
Okay.	0.52

MODELS

The model proposed draws inspiration from Li et al. [13]. Three main components play a role in the final dialogue generation network: a forward network **F**, a backward network **B**, and a final network **R** trained using Reinforcement Learning. The next sections will analyze each one of them individually.

Forward Network

This encoder-decoder network functions as a map between a message paired with a dialogue context and a response, similar to Sordoni et al. [22]. Therefore, our encoder receives as input the concatenation of the context and the input message.

Regarding the details of the implementation, GRU cells [4] are used, due to their relative simplicity compared to the more complicated LSTM cell [11]. A bidirectional RNN is used for the encoder as it has been shown that it is successful in similar tasks [2].

Backward Network

This network receives an utterance as input, and it has to predict the previous sentence that would have occurred in a natural-sounding dialogue. The same implementation details as for the Forward network are used. In both cases the cost function is the Cross-Entropy Loss.

Reinforcement Learning Network

The final network is trained using Policy Gradient optimization techniques. The policy is parameterized using a seq2seq model and its weights are initialized from the weights of the F network. Using examples from the dataset, training is achieved by performing a Monte-Carlo roll-out of one or multiple transitions according to the decoder policy. The REINFORCE algorithm is implemented together with a baseline value to reduce the variance that occurs while training. T, considering the dull utterances from Table 1 when calculating the reward score referred as *Ease of Answering* by the authors.

Rewards

The heuristically determined rewards used are the ones proposed by Li et al. [13]. The first reward (r_1) aims to drift the model away from generating responses that may lead the conversation towards dull sequences. In the formula below, S is a hardcoded list of generic responses, dependent on the dataset, that contains the most frequent target answers (e.g., *'I don't know'*), a is the response generated by the network, $N_S = |S|$, and N_s is the number of tokens in s.

$$r_1 = -\frac{1}{N_S}\sum_{s \in S}\frac{1}{N_s} log(p_F(s|a))$$

The conditional probability p_F is calculated using the pre-trained sequence-to-sequence network F. In our case, the hardcoded list of frequent responses is, also, presented in Table 1.

The second reward (r_2) penalizes the agent if it generates similar responses in consecutives turns. Thus, considering the dialogue A, B, C, we transform each of the sequences A and C into fixed vector representations, h_A and, respectively, h_C, through an encoder layer and then compute the logarithm of the cosine similarity of the two embeddings. We also add a threshold $e > 0$ to deal with the fact that the logarithm is defined only for positive values. In our experiments the value is set to $e = 10^{-10}$.

$$r_2 =- log(max(\frac{h_A \cdot h_C}{\|h_A\| \cdot \|h_B\|}, e))$$

The third reward (r_3) keeps the agent from diverging and generating unintelligible sequences by rewarding semantic coherent responses. Here we also consider the dialogue sequence A, B, C, with N_B the number of tokens in the sequence B, and N_C the number of tokens in C.

$$r_3 = \frac{1}{N_C} log(p_F(C|B,A)) + \frac{1}{N_B} log(p_B(B|C))$$

Given these three equations, the final reward, at the end of a transition, can be computed by:

$$R = \lambda_1 \cdot r_1 + \lambda_2 \cdot r_2 + \lambda_3 \cdot r_3 ,$$

where the coefficients suggested by the authors are $\lambda_1 = 0.25$, $\lambda_2 = 0.25$ and $\lambda_3 = 0.5$. We have experimented with different values for these coefficients as shown in the next section.

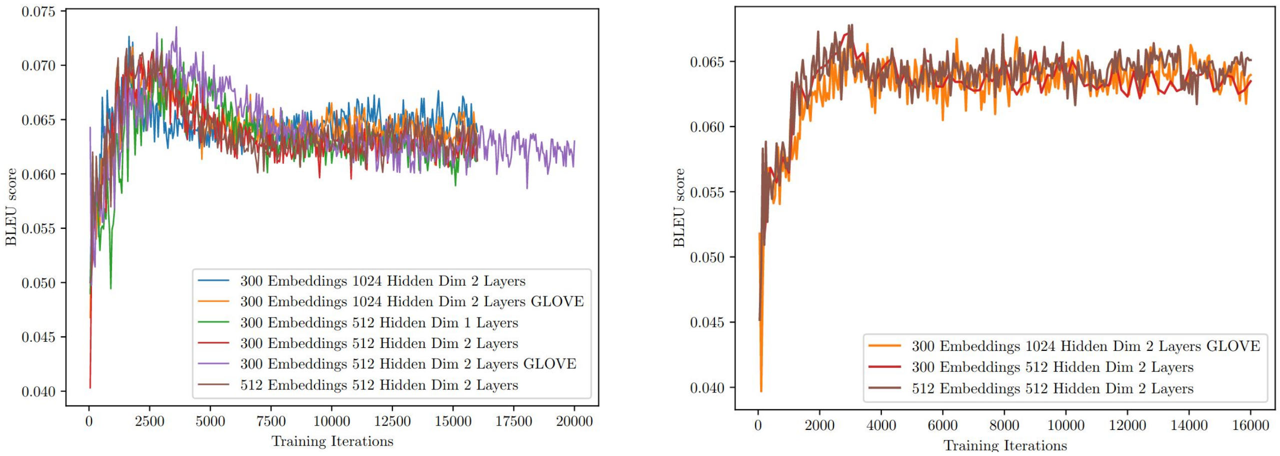
EXPERIMENTS AND RESULTS

One of the first decisions that we made in our experiments was to eliminate the context. Initially, training using one utterance as the context, we observed the inability of the model to generalize and generate relevant and coherent messages for a conversation that spanned multiple turns.

An example of this behaviour is shown in Table 2. This is caused by the small size of the training dataset and, therefore, the model's unpredictability when it receives unseen pairs of contexts and messages.

To evaluate our models we employ a diversity metric [12], representing the number of unique unigrams and bigrams generated normalized by the total number of tokens generated, to objectively measure the diversity of the responses. A higher diversity metric correlates to more diverse responses. Also, we make use of the Bilingual Evaluation Understudy (BLEU) [16] score to choose between different hyperparameters when training the Forward and the Backward networks. BLEU score is not a perfect measure, but it has a correlation with human judgment as shown by Galley et al. [9].

In Figure 2, the BLEU score is plotted for the validation dataset throughout the training steps.



a) Forward network **b) Backward network**

Figure 2. BLEU score while training for validation sets. An overfitting pattern can be observed

Both models appear to overfit, this phenomenon being more visible in the case of the Forward network. Inspecting the test data manually, we conclude that, if we allow the model to further learn, the responses tend to be more diverse and accurate, although some mismatches emerge. Some relevant examples are depicted in Table 3. The diversity metric [12] stands as another proof of this occurrence, showing an increase from 0.006 to 0.017 when unigrams are considered, and from 0.02 to 0.13, for bigrams. The first metrics are calculated before the overfitting occurs, while the second is computed after the process. We observe that diversity is no longer a problem after this pattern occurs, so the final network will only be used on the models whose training was stopped before the validation scores would have decreased. A similar observation is made by Csaky [6], who uses a Transformer model on the same dataset and also notes this behaviour.

Table 2. Example of the model degenerating. Once it outputs UNK, it stops generating meaningful responses

User: Hello.
Bot: Hello.
User: How are you?.
Bot: Fine.
User: What’s your name?
Bot: UNK.
User: How are you?.
Bot: I don’t know I don’t know.

We continue to train the model that suffers considerably from the diversity issue, the one before overfitting occurs on the BLEU score (0.071 for the Forward network and 0.069 for the Backward network), and apply Reinforcement Learning to improve its diversity.

Table 3. Comparison between the model before score decreasing and the one at convergence

Input	Before overfitting	At convergence
Hello.	Hi.	It’s me.
How old are you?	I don’t know.	Five.
How are you?	I don’t know.	Head still secure to the neck.

Mixed Incremental Cross-Entropy Reinforce (MIXER) [17] is utilized first with just the third reward, as suggested by Li et al. [13]. The R network is initialized with the parameters of the F network and then, we train for T - Δ steps in the same supervised fashion as before, and for the remaining Δ steps we use the REINFORCE algorithm. We increase gradually the value of Δ until all the sequence is trained with Reinforcement Learning. The diversity increases considerably from 0.006 to 0.032 for unigrams and from 0.017 to 0.144 for bigrams.

Following this, we train the model by using all the three rewards for 5 transitions per episode, setting the discount factor to 0 because no clear distinction could be made between a final and a non-final state. Following multiple experiments, the coefficients used for the rewards are changed to 1.0 for the first reward, 5.0 to the second reward and 0.1 to the third reward, to eliminate the model's tendency to converge to a safe space where it generates a single generic response to all inputs. These values are found through an empirical search, by observing that the third reward highly influences the agent and forces it to output the same response for multiple turns, while the second reward scarcely has an impact on the final model. Moreover, we changed the way the similarities of two sentences are computed in the second reward to use the embeddings and not the encoder final hidden states as suggested by Li et al. [13]. This is due the model's inability to generalize well to unseen messages sampled in the Monte-Carlo generation process.

The final diversities scores are shown in Table 4, where the F model is the model before overfitting occurs, the F_O model is the one after overfitting, R_M is the model after MIXER and R is the final model. The main observation is that MIXER and overfitting lead to the greatest relative increase, but the final model manages to achieve the best diversity scores.

Table 4. Final diversity scores

	Unigram	Bigram
F	0.006	0.017
F_O	0.027	0.137
R_M	0.032	0.144
R	**0.033**	**0.16**

In Table 5 a comparison is shown between our model and the model implemented by Li et al. [13]. The responses for their model are taken explicitly from their paper. Both models offer diverse and relevant responses, but from these examples one can observe that their model is more interactive, as it asks more questions, due to its exposure to more data and epochs for learning.

Table 5. Final responses

Input	R	Chatbot [13]
How old are you?	Twenty eight.	I'm 16, why are you asking?
What's your full name?	Roy.	What's yours?
How much time do you have here?	Not enough. What do you want?	Ten seconds.

CONCLUSIONS

The experiments conducted in this paper have shown the ability of Reinforcement Learning to allow the model to deflect from the diversity issue. This approach is valid even for a smaller dataset with short sentences as the one used in our research.

The main observation is that the model trained in a supervised fashion, using the standard Cross Entropy Loss, suffers considerably from the diversity issue. This problem can be alleviated by allowing the model to overfit on the training dataset, but actually the best results appear after Reinforcement Learning is applied.

The limitation of the final chatbot comes from the fact that, being trained on a small dataset, the agent cannot perform a coherent and consistent conversation that spawns more than a few turns. That being said, the results achieved in this paper are significant with respect to future research in exploring other variants of rewards, datasets or architectures combined with Reinforcement Learning. This kind of empirical research is beneficial in understanding the capabilities of the neural networks employed for building deep learning chatbots.

REFERENCES

1. Adiwardana, D., Luong, M. T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., ... & Le, Q. V. (2020). Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977.

2. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

3. Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In Advances in Neural Information Processing Systems (pp. 1171-1179)

4. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

5. Cleverbot https://www.cleverbot.com/. Accessed May 20, 2020.

6. Csaky, R. (2019). Deep learning based chatbot models. arXiv preprint arXiv:1908.08835.

7. Danescu-Niculescu-Mizil, C., & Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. arXiv preprint arXiv:1106.3077.

8. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

9. Galley, M., Brockett, C., Sordoni, A., Ji, Y., Auli, M., Quirk, C., ... & Dolan, B. (2015). deltableu: A

discriminative metric for generation tasks with intrinsically diverse targets. arXiv preprint arXiv:1506.06863.

10. Grudin, J., & Jacques, R. (2019, May). Chatbots, humbots, and the quest for artificial general intelligence. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (pp. 1-11). DOI:https://doi.org/10.1145/3290605.3300439

11. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

12. Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2015). A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055.

13. Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., & Jurafsky, D. (2016). Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:1606.01541.

14. Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., & Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547.

15. Lison, P., & Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

16. Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

17. Ranzato, M. A., Chopra, S., Auli, M., & Zaremba, W. (2015). Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732.

18. Ritter, A., Cherry, C., & Dolan, W. B. (2011, July). Data-driven response generation in social media. Proc. Conference on Empirical Methods in Natural Language Processing 2011 (pp. 583-593).

19. Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016, March). Building end-to-end dialogue systems using generative hierarchical neural network models. In Thirtieth AAAI Conference on Artificial Intelligence.

20. Serban, I. V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., & Bengio, Y. (2017, February). A hierarchical latent variable encoder-decoder model for generating dialogues. In Thirty-First AAAI Conference on Artificial Intelligence.

21. Shang, L., Lu, Z., & Li, H. (2015). Neural responding machine for short-text conversation. arXiv preprint arXiv:1503.02364.

22. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., ... & Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. arXiv preprint arXiv:1506.06714.

23. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems(pp. 5998-6008).

25. Vinyals, O., & Le, Q. (2015). A neural conversational model. arXiv preprint arXiv:1506.05869.

26. Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. Neural computation, 1(2), 270-280.

27. Zhou, L., Gao, J., Li, D., & Shum, H. Y. (2020). The design and implementation of xiaoice, an empathetic social chatbot. Computational Linguistics, 46(1), 53-93.

Design features of a VR software system for personnel training in aviation

Andrei Bulai, Diana Andronache, Dorin-Mircea Popovici

Ovidius University of Constanta

124 Mamaia Bd, 900527, Constanta Romania

bulai.andrei10@gmail.com; diana.andronache7@gmail.com; dmpopovici@univ-ovidius.ro

DOI: 10.37789/rochi.2020.1.1.20

ABSTRACT

This paper presents the steps that must be taken to develop a Simulator type software using VR technology and the presentation of our own software system for the training aviation personnel. We present the state of the art of VR and describe the developed application through the chosen solution, meeting all the criteria of a learning and testing environment for the aviation personnel, simulating the targeting operations of a helicopter and adjacent elements. In fact, we make a description of both the data structure and the technical design. Logical design and system architecture, use cases and conceptual classes are also presented. The technical part describes in detail the implementation stages of the application.

Author Keywords

Virtual reality; Virtual Simulation; Software development; Triggers and Gestures; User Interface for virtual reality; User Experience for virtual reality; 3D animation and modeling

ACM Classification Keywords

H.4.m. Miscellaneous; H.1.2. User/Machine Systems; D.2.m. Miscellaneous; D.2.10 Design; I.3.m. Miscellaneous; I.3.8; I.3.7; I.3.6; I.3.5

General Terms

Human Factors; Design; Measurement.

INTRODUCTION

At the moment, VR technology has gone from the early stage to the expansion stage to a small-scale on a large scale. The concept is well known, and the main equipment manufacturers that are part of VR systems are investing resources in the development and improvement of new technology. Therefore, today, the past VR engineering ideas can be restructured and homogenized both with the new market requirements and with the new equipment. This is why we afford to accept the challenge of developing a relatively low-cost training virtual environment dedicated to aviation personel.

Predecesors

In terms of aviation, since the first flight simulator in 1966, developed by military engineer Thomas Furness for the US Air Force [1], in a project to present the concept of VR, the market has evolved very quickly. Thus today we have the opportunity to use professional simulators both in terms of the complexity of simulating the real aviation environment and in terms of graphic quality.

Next we present the state of the art in the field of flight simulators, stating some of the most relevant projects launched on the market.

Microsoft Flight Simulator X – is a simulator produced by Aces Game Studio, being among the first VR simulators with advanced graphics. The first versions appeared in 2006, on the Microsoft Windows platform. It is constantly evolving, with new concepts being implemented with each update [2].

DCS (Digital Combat Simulator) World – is a simulator produced by Eagle Dynamics, the first version was released in 2008, the current version can be obtained for free from the Steam platform. It is a simulator dedicated to the military environment, with users having countless aircraft models and more. The focus is on specific operations and missions [3].

Aerofly FS 2 Flight Simulator – is another flight simulator with an emphasis on photorealistic scenarios, produced by IPACS, running on the platforms: Android, MAC, Windows and iOS. It was launched in 2014, and users can enjoy a global elevation in image, of over 300 handmade airports in West America, 3D construction, bridges, highways and detailed cities. In terms of aircraft models, the detail is impressive, the equipment and tools in the cockpit being animated, which gives a high level of interaction with the environment. The implementation of the physical elements is performed with great accuracy, and together with the fluidity of the frames with which they managed to display the image, they created an even stronger immersion effect of the user. Other navigation functions that we can find are: Route planning, ILS (Instrument Landing System), VOR (Omni Directional Radio Range) and NDB (Non-directional Radio Beacon). Moreover, this simulator provides intuitive support for Oculus Rift and HTC Vive, without any additional software [4].

X Plane 11 – it is perhaps the best rated simulator on the market at the moment. The first version appeared on November 25, 2016, and the most current on December 12, 2019. This simulator, also perceived as a video game, runs on the platforms: Windows, MAC and Linux. Users can enjoy similar graphics and experience to reality, enjoying an impressive number of airports: 13.000 [5].

Virtual Marshalling Simulator – is an stand-alone training system produced by Virtual Simulation Systems used by the Royal Australian Air Force, the Royal Australian Navy, and Australian Army Aviation. It is specializes for ground based support crew, offering a high fidelity immersion into the world of flight deck operations with the ability to control and direct aircraft that are about to deploy or have returned from a mission and require ground marshalling. The software has dynamic weather effects with a range of settings, including wind, rain, fog, lightning, real-time scenario editor with vast control, ability to trigger aircraft emergencies such as engine fires, hydraulic leaks, hot brakes on fighter planes, and tarmac incursions by personnel or vehicles, high-fidelity virtual simulation with authentic graphic, sound and more. Simulated procedures include pilot signals, refueling, deck lashing, power, take-off and landing, and many others. The main purpose of the system is to reduce the rate of effort on already scarce resources, reliance on pilot availability and expenditure on fuel and other consumables. Otherwise reduces logistic setup for FARP (Forward Arming and Refueling Point) marshalling exercises [6].

A VR-BASED TRAINING SYSTEM COMPONENTS

The structure of a VR system, which aims at the interaction of the user subjected to the training or testing process we explain it in the fallowing way (Figure 1).

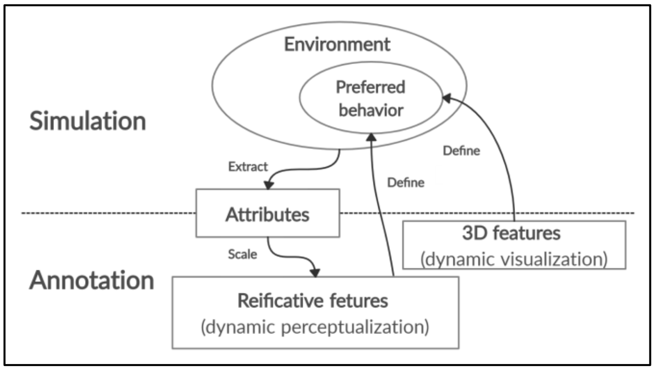


Figure 1 - Annotation of interactive simulation. Adaptation of [7].

Preferred behaviour of the simulated environment is annotated as 3D and reificative features [7]. The annotation action is one of the most natural ways for human beings to analyze and interact with documents, images and different objects [8]. Depending on the nature of the objects and events that take place in the simulated environment, the attributes of the simulation play the role of scaling factors on the reifying characteristics. In our case, a gesture or a movement of the user's hand will automatically adapt to the length of each user's arm. 3D functions are naturally calibrated to a correct scale, without modifications. In fact, 3D features can only be perceived.

A VR system consists of four main components. Dedicated software or engine for producing and managing graphics, Tracking system that constantly knows where the user or users are on stage, the Hardware part that allows Video and Audio viewing of the 3D scene and, most importantly, data content 3D.

Dedicated software or engine has several tasks to perform. First, it is the one that produces or takes over the 3D graphics. In addition, it is the one that takes the input information from the input and tracking devices and also provides the platform or environment in which the 3D scene can be developed.

The tracking system determines the position of users in the virtual world and usually uses a camera with tracking sensors that record movement. In the case of Vive or Oculus, these tracking sensors are provided as part of the HMD (Head-Mounted Display) package, but not every HMD model has this integrated part.

The visualization system represents the Hardware part, which allows the visualization of the 3D virtual scene in stereo. In the case of Vive or Oculus, this is also done via the HMD.

The 3D data or the 3D scene itself makes it possible to view and interact the user in VR. This data is the basic resource of a VR environment. If we are talking about a game, then the 3D data represents both their models and animations, as well as the background code responsible for the user's interaction with them. All the functionalities within an application, as well as the models that take part in the action are the primary resource in a VR System.

In fact, in addition to these components, the user is also part of the whole virtual environment [9]. Virtual objects are subjects in the users' direct or indirect interactions and may enhance collaboration between users. In other words, the virtual space must be constructed, first of all, considering the user's cognitive and empirical attributes. When we create virtual space models, the base criterion should be the accuracy of the human representation of reality which may not necessarily correspond with reality. To this end, the human experience is first constructed by situating the user in the virtual context, then tested through the user's direct interaction with the environment, and reconsidered, in a recursive process [10].

OUR PROPOSED VR SYSTEM FOR PERSONAL TRAINING IN AVIATION

In the following we present the context of choosing the solution, the stages we went through in the realization of the software system, as well as the technical exemplification of the important processes. We chose to make this application after a visit to Tuzla International Airport, where after some discussions with the management team, we decided that an optimal scenario, out of several possible ones, of a software product would be the ground work of the aviation personnel. The development on the part of VR in terms of flight simulation is growing. From this point of view, but also because such a project involves a fairly large and well-trained team with specialists in the field (pilots, aeronautical engineers and so on).

We chose as main objective to focus on the activities of staff in the field and more precisely, of those activities that have not been approached by the production studios until now.

In addition, we chose to create a unique setting, the location and context of the scenario being a real one (Cliff of the Casino in Constanța, Romania).

The application meets all the conditions of a learning and testing environment for aviation personnel, so it can simulate the aiming operations of a helicopter and the adjacent tasks.

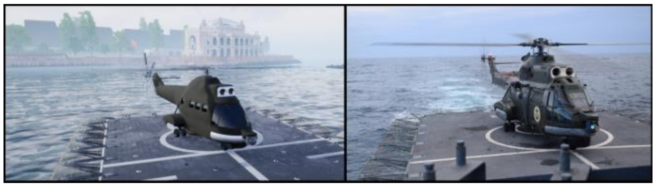


Figure 2 - Models in the foreground

The final version of the application contains the complete modeling of the chosen scenario and the general environment: background models, complete and complex models in the foreground (the flight deck of a military frigate model T22, a helicopter model Puma Naval - IAR 330 and the Casino) (Figure 2).

LOGICAL DESIGN

In view of the work plan, the next step is logical design. Here we have made decisions regarding the application architecture, technologies and concepts that we will continue to use and most importantly the use cases of the software system.

Architecture up to a certain point is an art, but from a certain point it is a science. To build something real, which will withstand the time factor requires knowledge. We preferred to give extra time to the minor details, but at the same time to keep the simplicity of the product, without detailing unnecessary parts that do not have their purpose in the application.

We chose a minimalist, simplistic, suggestive design, which does nothing but emphasize the important parts of the simulated environment, just to eliminate all the risks of the development plan, so all use cases can be achievable and the entropy factor software should not appear in future versions.

In addition to the general environment, which involves the sea plan, the geographical plan, the SkyLight and the Skybox, using Blueprints (visual scripting technology provided by Unreal Engine), we created: materials, objects, object instances, all with unique properties for each one.

Following a taxonomy process applied to the “Helicopter” concept, we developed the fully functional model / asset of the IAR-330 helicopter. Its animation is activated sequentially by triggers and gestures.

The connection of HMD to the application is done both by Blueprints and by control classes (C++). The dedicated physical aspects of the simulated context were also implemented by coding.

At the level of user interaction, friendly communication techniques were addressed, such as: assisted execution of tasks through instructions, help tools, dialogue, sound and relevant and intuitive noises. In addition, the application contains a menu that fully covers the range of use cases.

Use cases

Following an analysis of the scenario, we made a series of use cases:

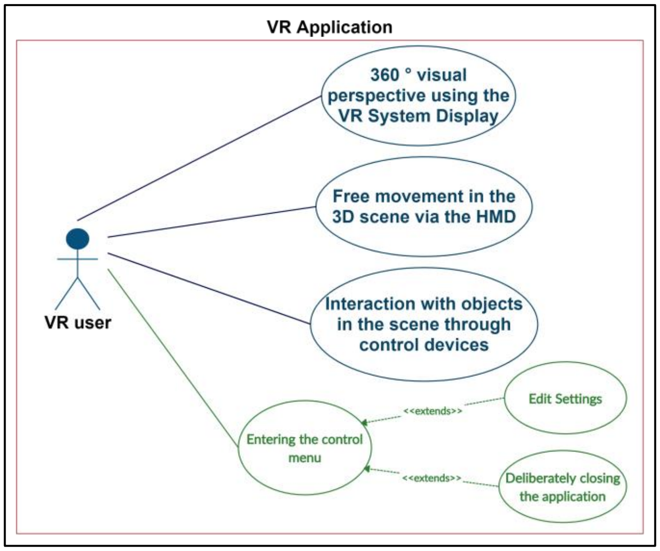

VR Application
360 ° visual perspective using the VR System Display
Free movement in the 3D scene via the HMD
Interaction with objects in the scene through control devices
VR user
Edit Settings
<<extends>>
Entering the control menu
<<extends>>
Deliberately closing the application

Figure 3 – Use cases Diagram

We identified the main characteristics of the actor (unique in this case), this being the user, based on a plan to simulate communications with the system (Figure 3). The use cases do nothing but model the system as desired from the end user's point of view.

They clearly describe how the user will interact with the system, then provide us with a basis for testing the status of the application. Of course, there may be multiple users of the application, but for the system they all play the same role.

First, the system must provide the user with three essential, basic things of the simulated framework, which are:
- 360 degree visual perspective, through the display provided by the VR System.
- Free movement in the 3D scene via the HMD or other tracking system.
- Interaction with objects in the scene through control devices and provoked events.

In addition to these use cases that achieve the purpose of the application itself, to provide a simulated reality environment for aviation personnel, there are others, which are part of any other system of interaction with a human actor through an interface like entering the main menu and the control menu, enterering the editing mode of the settings wich is an exceptional behavior, being optional in addition to the previous use case and completion of the current process and exit from the application wich is also an exceptional behavior.

Conceptual classes

To identify the conceptual classes of the context of the problem, we applied a grammatical analysis on the description of the functioning of the system.

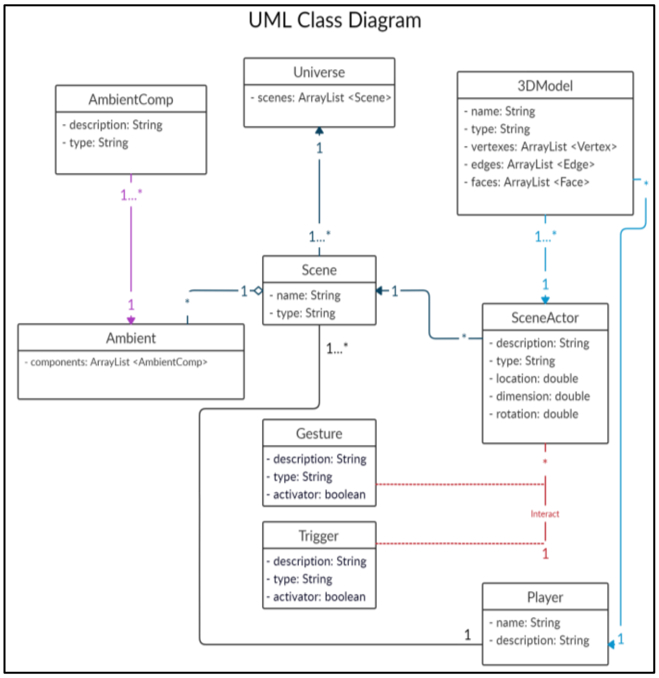


Figure 4 – Conceptual class Diagram

Thus, nouns become potential objects, the classes being identified through the following criteria [13]:

• retained information - the data retained by the object is important for the functionality of the system.

• necessary services - the object must have the ability to change the value of its attributes in a certain way through identifiable sets of operations.

• multiple attributes - objects with a single attribute can be better represented as attributes of other objects.

• common attributes - object attribute sets apply to all instances of this object.

• joint operations - sets of object operations apply to all instances of this object.

• essential requirements - external entities that consume information that is essential for the proper functioning of any system solution, will almost always be defined as objects in the requirements model.

A potential candidate should meet all these selection characteristics in order to be included in our model.

The class diagram (Figure 4) is made in a conceptual perspective, but even more so when we talk about a software product made in an engine like Unreal Engine, where, in the case of a single project, hundreds or maybe even thousands of unique objects can be created. Indeed, these objects have common attributes and can be correctly framed under the definition of the same general concept, the language of the field being captured much faster. All of our 3D assets models are instances of objects represented by the conceptual class defined Model3D, which has common conceptual attributes.

Data structure

The first step in carrying out the project was to gather as much useful information as possible. For a good understanding of the problem, first, we needed a good detailing of work tools. Both in concept (blackbox) and in detail (where it is vital).

Unreal Engine is the software engine through which we were able to develop our system. It is also developed by Epic Games, first introduced in 1998.

Unreal Engine 4.24 [11] was the engine version available when we started developing the application, and Unreal Engine 4.25 is the latest stable version released by Epic Games so far.

The visual system of Blueprints provided is a complete system of scripts, based on nodes and graphical interfaces to create game elements inside the editor. As in many other common scripting languages, it is used to define object-oriented (OO) classes or objects [12].

This system is extremely flexible, being dedicated to both designers and programmers, either for its practical utility, which provides you with the full range of concepts and tools that normally only programmers have at their disposal, or for its ability to create basic systems in C++ that can be extended by designers.

Although they are called matrices, blueprints are actually lists. Structures are used to make complex data. It is preferable to avoid creating structures within other structures, in favor of creating basic methods. In fact, the Epic Games team's own architecture is specially designed to facilitate both the effort of programmers and the work of designers in creating projects.

TECHNICAL DESIGN

In view of the work plan, the next step is the technical design. Now that we have gathered the necessary materials and laid the foundations for the construction of the application, we can move on to implementation.

Communication of the user through triggers and gestures

When we started to put ideas together at a conceptual level, we set out to make the transition as easy as possible from what classical software systems engineering involves, to the new requirements that come in approaching a software system using VR technology. Thus, in addition to the usefulness of triggers, from classic applications, we also used gesture recognition.

A trigger activates an event or series of events. If we are talking about a movie sequence or an animation, then it is very possible that we want to activate it in a special context. Getting a reference to the sequence we made, we created a Blueprint which in turn generated a C++ code sequence that we can set so that it starts and stops when we want it to.

Regarding the sequential activation of the animation, we chose to do it by implementing triggers and gestures.

Thus, in addition to use cases and conceptual classes, we can complete the basic architecture with the sequence diagram (Figure 6).

Before the image reaches the display, triggers are activated, which in turn activate functions for gesture recognition. If the gestures correspond to gestures already saved, then an ID will be used to activate an animation sequence. Thus, only now, the provoked events can be visualized (Figure 6).

Helicopter movement animation can be controlled by the ground fly deck personnel support using hand signals such as: direction correction, landing and deck lashing.

There are several types of gestures which we also used for this application, so they fall into several categories: directional movement gestures, flow control gestures, spatial orientation gestures, multifunctional gestures (which can trigger multiple events) and tactile gestures [14].

The events caused by classic gestures activated by the buttons of the keyboard, mouse or even VR controllers are the flick, the pinch and rotation.

Based on them, their combinations, but also other imports of gestures registered by dedicated systems, an almost unlimited software implementation can be reached; the only limitations being those of physical laws in reality.

We chose to work with components dedicated to VR systems by MotionController. Using the MotionController Blueprints we added a gesture tracking component and four functions that work in parallel:

- the start function of the gesture recording
- the end function of the gesture recording
- gesture recognition start function
- the end function of gesture recognition

The first two functions are related to a gesture recording input action, which more precisely, begin to record the gesture performed by the sensors. The next two are linked to a trigger recognition action for gesture recognition which verifies if the gestures recorded by the sensors are valid or not (Figure 5). The recognized result of the gesture has an ID, which will help us in establishing the event followed by the correct recognition of each gesture.

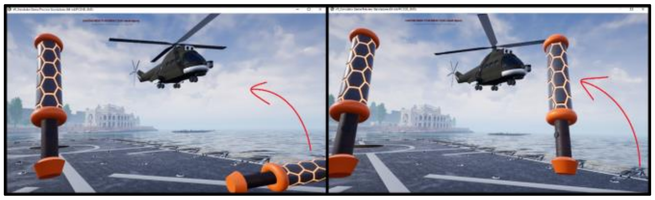


Figure 5 - Gesture recording by activating a trigger

For the space tracking functionality of the gestures, we will also use a tracking component, which we will connect to a space drawing function and another one for predicting the drawn gesture.

If a continuous recognition of gestures in space is desired, without this taking place during the pressing of a trigger (from the controller), although in this case it is not preferable, an input action of their continuous recording will be used.

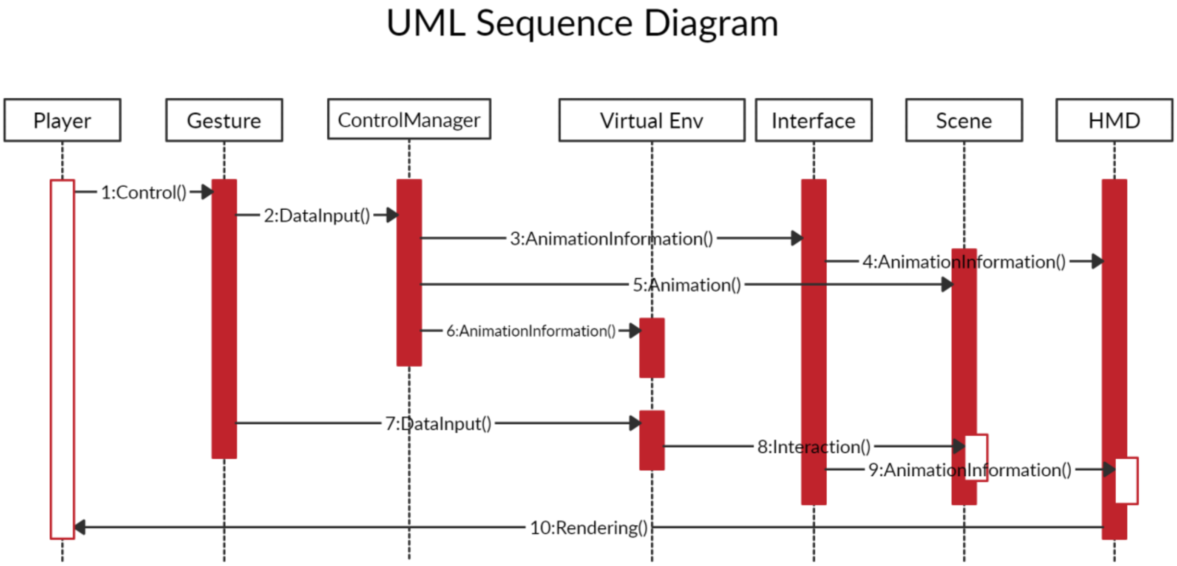


Figure 6 – Sequence diagram

Obviously, even if we had approached the continuous recognition of gestures, activating this trigger from the application at an inappropriate time, it would not have produced an event, because the animation sequence of the helicopter would not have ended. Therefore, both by recognizing by means of triggers (by pressing a button on the controller) and by continuous recognition (of the attempt to permanently recognize gestures), the helicopter will first finish its "process" started.

From the beginning of the development of the application, we started from the idea that the final product should look as simple as possible, but with a greater graphic impact. Therefore, we kept the minimalist concepts of simplicity in terms of user interaction with the system.

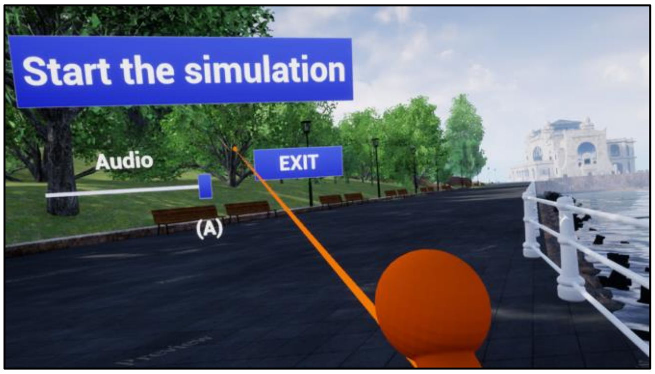


Figure 7 - Perspective from the first scene

Conceptually speaking, the universe within the application is made up of a single scene. However, technically we use the same scene, duplicated to give the user a double perspective in the created universe (Figure 7). The menu follows the user at every moment within the environment (Figure 8). It consists of a simple panel with two buttons: one that makes the transition of the scenes, in our case from scene A to scene B, but also from scene B to scene A, and the second button makes it possible to close assisted, secure application.

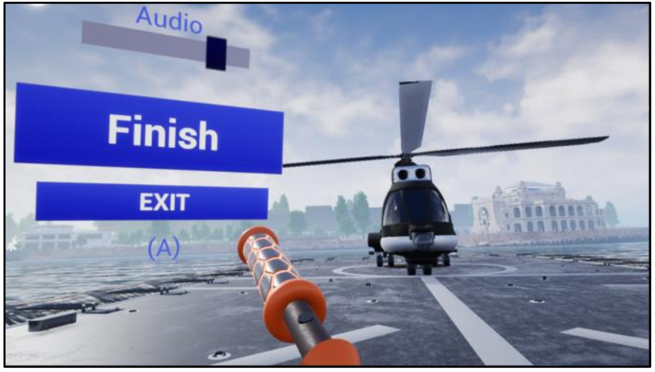


Figure 8 – Perspective from the second scene

The menu provides intuitive usage information, communicating to the user by its nature, but also by text indications. In addition, it is possible to edit the settings by changing the parameters. The simple and friendly aspect of the interface was made with the help of the graphical editing mode, familiar to the one from Visual Studio.

The application can be used via any HMD, we using Oculus Rift S. The hardware configuration of the HMD was done through functions implemented using Blueprint technology.

The audio element is vital in the composition of any final product of this type. The user lives the visual experience differently if it is introduced in the virtual environment and through a stereo system. In terms of sound, we created a main class and a sound mixer. All sound effects used in the application have been introduced in the parent class.

To add sound effects to the scene, we used replicas of the sounds, placed as sub-objects of the actors in the scene. It should be noted that a sound replica does not necessarily have to be related to an existing object in the scene. It can also be introduced as a singular element.

Modeling

The scene represents the space and place where the user is at a given time. This is part of the universe of a game or, in this case, the simulator, as defined in the class diagram. The scenes are populated with actors, and the actors, in turn, are made up of one or more 3D models. In a project with complex graphic content, modeling is not done directly by handwriting code. Instead, automatic code generation techniques are used, similar to the Blueprint graphical visualization, through which methods are used that contain methods, which in turn write code automatically. Developers can later access the C++ code of those classes, and can make changes or troubleshooting in case of unstable versions or plug-ins.

The Unreal modeling plug-in offers functions similar to those in dedicated software. With the help of modeling techniques such as: NURBS (Non-Uniform Rational Basis Spline) modeling, polygonal modeling and NURMS (Non-Uniform Rational Mesh Smooth) modeling, we managed to reach results similar to those obtained in professional applications.

Animation

Once the 3D models of all the actors are created, we can move on to their animation processes (where appropriate). Perhaps the most rudimentary form of animation is that of the material applied itself. In the case of a larger animation, each mesh to be manipulated in different ways must be analyzed separately. In this case, we could use the taxonomy created on the model of the IAR 330 Puma Naval helicopter, applying animations on each essential element. Thus, from understanding the main components, we were able to move more easily to understanding in detail, at a more granular level of the concept. So we decided on the type of approach we would take in the separate animation of the main rotor assembly, the tail rotor assembly, and the entire helicopter body.

At the scenario level, we have developed a series of possibilities for the trajectories of the helicopter to the point of control of its animation by the user (Figure 9).

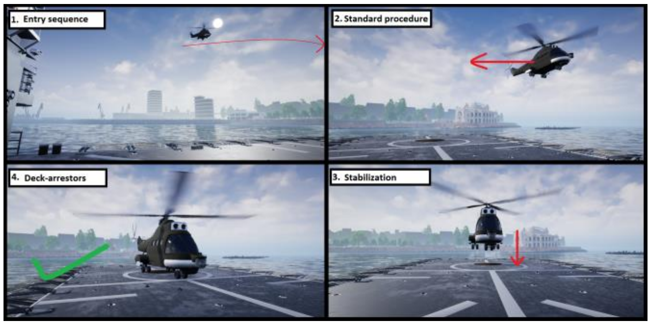


Figure 9 – Animation script

The next step in the animation process was the actual animation of the helicopter parts. Due to the nature of the 3D Model studied, there was no need to go through the process of Rigging and Skinning, but we needed again the taxonomy created in the evaluation of the rotor parts (Figure 10).

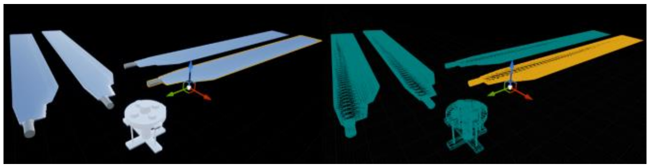


Figure 10 - Parts of the helicopter to be animated based on the taxonomy

The rotation function of an object is implemented via the RotatingMovementComponent class in the standard GameFramework Unreal Engine package. This function can be added to the actors of a scene and can be used for different purposes. In principle, if the pivot of the animated object is not changed, then the composition of the new location is done by default through it.

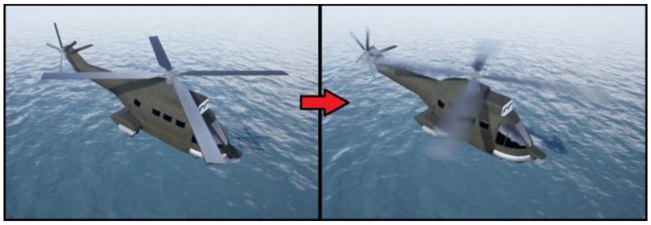


Figure 11 - Animation of the main rotor and the tail rotor together with their propellers

Thus, using the rotational component of the movement on the main rotor block and the secondary rotor block (tail) we were able to animate the essential components in question of the helicopter (Figure 11).

Both in the elaboration of the animation script and in the placement of the actors on stage we focused on: the correct lighting of the scene and its interaction with the materials of the 3D models, the positioning of the camera and the user's perspective to simulate the immersion in the virtual environment and balancing the speed at which the animation happens in the development phase with the number of frames processed per second of the VR system. Given that such a display does not have the ability to process a very high number of frames per second, we had to make a compromise in terms of the actual rotational speed of the helicopter blades. Thus, on the display of the VR system you can perceive as naturally as possible their rotational movement.

At the level of composition, we recreated the panorama of the real cliff scenario, using textured materials in the same style, thus completely homogenizing the chosen context. In addition, we used techniques specific to creating niche games and graphics applications. Thus, in order to keep a better optimization of the data flow processed by the system and to obtain as many FPS (frames per second) as possible, we opted in the detailed modeling of the foreground elements and the important elements in image register, leaving the background populated with texturally poor objects, but especially polygonal, for a faster rendering of the image.

In terms of animation, at least half of the effect created in an application is sound. Although the process of creating images itself is more expensive, the emphasis should be equally on the audio side. Most of the time, as in the case of film production, a good image combined with poor sound quality is much worse than a bad image, but it comes with a high sound quality. The audio part can save the final product from failure or throw it even harder into the abyss.

For the sound caused by the engine, the rotor and the blades of the helicopter we used an audio material in wav format.

The last step we took in finalizing the animation was the final editing and the general aspect of the image-sound fluidity. Once we changed certain parameters that were not exactly in place, through a subjective analysis of duality we arrived at the desired final version of the animation of the main actors.

CONCLUSION

Following the stages of content creation in a logical order of the modeling and animation processes, but also of the texturing and material creation processes, we managed to create the basic component of a VR system. By understanding the related concepts and technologies we were able to connect the created component to the tracking and viewing components provided by the HMD Oculus Rift S device together with the Oculus touch devices. Thus, adding to the equation the Unreal Engine and the processing hardware, we created the complete framework of the VR System infrastructure.

In designing the use cases, we approached a realistic attitude, calculating all the variants and risks of the development process. Thus, we did not intend to implement use cases, which require a higher level of experience, precisely in order to finally reach a final product, with its positive parts, even if it still has limitations. Their technical design and implementation were the main learning ingredient. During the implementation there were times when we had to make decisions, with varying degrees of importance. After making the wrong decisions, we were able to go back and learn from them. Moreover, through testing sessions, we redesigned some of the basic architecture so that we can achieve better results. Our user perspective has helped us fix most of the issues that have arisen during implementation.

The software system fully responds to the desired and proposed appearance issues. However, both its nature and the technologies used lead to a list of limitations that can be reduced by later versions of the application or other interpretations as the technology evolves. First, the nature of the simulated virtual environment will never be compared to reality. The user's immersion in the virtual environment will never be equivalent to that in reality, without direct interventions from the outside (for example: the underwater environment). Second, we are often limited by the space we are in.

As same as Virtual Marshalling Simulator, the application offers the advantages of learning processes, avoiding potential risks (damage 0 after the learning or training process), can preview a candidate's behavior following events, but can also serve for entertainment purposes.

The expansion of the software system in later versions may include more complex general visual areas, such as a systematic database used in the customization of tools and meshes, but also the choice of more complex scenario and more diversified simulation operations.

ACKNOWLEDGMENTS

Thanks go to the CeRVA team from the "Ovidius" University of Constanta. We also thank the management team of Tuzla Aerodrome in Constanta with special mentions to Regional Air Services.

REFERENCES

1. Leslie Mertz, *Virtual Reality Pioneer Tom Furness on the Past, Present, and Future of VR in Health Care*, Publisher: IEEE Pulse 10(3):9-11, 05 2019, ISBN: 2154-2287, DOI: 10.1109/MPULS.2019.2911808.

2. Cristiano Rodrigues, Rosaldo J.F. Rossetti, Daniel Castro Silva, Eugénio Oliveira, *Distributed flight simulation environment using flight simulator X*, Published in CISTI journal, 07 2015, DOI: 10.1109/CISTI.2015.7170615.

3. Digital combat simulator homepage, https://www.digitalcombatsimulator.com/en/.

4. Roganov V.R., Roganova E.V., Micheev M.J., Kuvshinova O.A., Zhashkova T.V., Gushchin S.M., *Flight simulator information support*, Published in journal: Defence S and T Technical Bulletin, 01 2018, ISBN: 1985-6571.

5. Fernando Soares Carnevale Ito, Roberto Santos Inoue, Luiz Carlos Querino Filho, Kalinka Castelo Branco, *Cooperative UAV formation control simulated in X-plane*, conference: ICUAS, 06 2017, DOI: 10.1109/ICUAS.2017.7991421.

6. Virtual Simulation Systems: Virtual Marshalling Simulator homepage, https://virtualsimulationsystems.com/site/index.php/simulation/fixed-wing/aircraft-marshalling.

7. Mikko J. Rissanen, Yoshihiro Kuroda, Tomohiro Kuroda, Hiroyuki Yoshihara, *A Novel Interface for Simulator Training: Describing and Presenting Manipulation Skill Through VR Annotations*, conference: HCI 07 2007, ISSN: 0302-9743, DOI: 10.1007/978-3-540-73335-5_57.

8. Stefanut T., Gorgan D., *Graphical annotation based interactive techniques in eTrace eLearning environment. "eLearning and Software for Education"*, eLSE 2008, The 4th International Scientific Conference, Ed. Universitara, ISBN: 978-973-749-362-0.

9. Popovici D.M., *A foray into 3D virtual environments (in romanian – O incursiune în mediile virtuale 3D)*, Ed. Muntenia, 2007, ISBN: 978-973-692-191-9.

10. Popovici D.M., Hamza-Lup F.G., Polceanu M., Zagan R., Gerval J.P., Querrec R., *3D Virtual Spaces Supporting Engineering Learning Activities*, Published in IJCCC, 11 2009, DOI: 10.15837/ijccc.2009.4.2456.

11. Paul Oliver, *Unreal Engine 4 Elemental*, 08 2012, DOI: 10.1145/23411836.2341909.

12. David Nixon, *Beginning Unreal Game Development. Foundation for Simple to Complex Games Using Unreal Engine 4*, Publisher: Apress, 02 2020, ISBN: 978-1-4842-5639-8 DOI: 10.1007/978-1-4842-5639-8_5.

13. Bogdan C., *Concern-oriented and ontology-based Modular Architectural Design of Software Systems*, 1st International Conference on Economics and Information Technology e-Society Knowleadge and Innovation, Bucharest, Romania, 2008, ISBN: 978-973-749-491-7.

14. Kasper Rise, Ole Andreas Alsos, *Human-Computer Interaction. Multimodal and Natural Interaction: Gesture-Based Interaction: Visual Gesture Mapping*, conference: HCI 07 2020, ISBN: 978-3-030-49061-4, DOI:10.1007/978-3-030-49062.

Usability Testing of Mobile Augmented Reality Applications for Cultural Heritage – A Systematic Literature Review

Diana Tiriteu, Silviu Vert

Communications Department, Politehnica University of Timișoara

Piata Victoriei No. 2, Timișoara, Romania

diana.tiriteu@student.upt.ro, silviu.vert@upt.ro

DOI: 10.37789/rochi.2020.1.1.21

ABSTRACT

This paper presents an overview of the existing literature to date regarding the usability testing of Mobile Augmented Reality applications developed for cultural heritage. A Systematic Literature Review was conducted with this purpose. Four databases were interrogated with specific keywords. Our of the 88 found research papers, only 31 met the eligibility criteria to be included in this survey. Four major usability testing methods were identified: questionnaires, focus groups, user testing and interviews. The questionnaire method was used in almost all of the studies, and only a few of them combined more usability testing methods. Nearly all papers targeted the outdoors use centered on location-based augmented reality. The perceived ease of use, usefulness and enjoyment were confirmed by a majority of the papers. The ease of use was also correlated to the ease of learning, as well as with the interest of using the application. Cultural heritage is one of the domains that can benefit greatly from the latest improvements in mobile augmented reality technologies.

Author Keywords

Mobile augmented reality; usability; user experience; questionnaire; focus group; user testing; interview; cultural heritage.

ACM Classification Keywords

• General and reference~Document types~Surveys and overviews • Human-centered computing~Human computer interaction (HCI)~Interaction paradigms~Mixed / augmented reality • Human-centered computing ~ Human computer interaction (HCI) ~ HCI design and evaluation methods ~ Usability testing

INTRODUCTION

This paper focuses on the usability testing of Mobile Augmented Reality (MAR) when applied for cultural heritage. In the past few years, cutting-edge technology has developed tremendously in smartphones, making it more approachable for all people. Along with this, application developers took advantage of this new fast-growing applicable domain and started creating apps for almost everything. Amongst those, Augmented Reality (AR) has grown in the mobile applications area, having now the widest audience ever. Together with application developers, cultural institutions are benefiting from this technology growth by trying to design and implement applications which support the cultural heritage [3]. And what better technology to fit their needs than AR? However, just having an app that seems to fit your need does not mean that the people using it are having the best user experience possible. Here, an important aspect is represented by the usability of the app.

User Experience (UX) covers all aspects of the end-user's interaction with an application. In a proposed model by Lee [23], UX is a combination of *usability* (perceived usefulness, ease of use and enjoyment) and personal characteristics. Considering this, the usability of an application will be high if the users will find the app easy to use, efficient and enjoyable.

Augmented Reality (AR)

Augmented Reality is a technology used to enhance the context of the real world. It works by overlaying digital representations on the objects from the physical world. Doing so, it enriches the user's perceptions on seeing, hearing, and feeling. All of this can be accomplished with the camera lens of a smartphone or tablet. A good AR design makes the virtual and the physical world coexist harmoniously, the user getting to perceive the overlaid AR elements as being part of the physical world. There are four different categories of AR technologies which are well described by Vanessa Camilleri in [36], and all of them are adaptable to different contexts:

Marker-based AR uses virtual markers that, when sensed by the reader (e.g. the mobile phone's camera), trigger a result. This type of AR uses the image recognition technology, and one of the simplest markers can be found in QR codes.

Markerless AR uses location-based technologies or GPS to provide data based on the location of the mobile device. The accelerometer embedded in smartphones provides the location coordinates for the app and activates the AR data.

This type of AR is mostly used in tourism and marketing apps, as it makes use of the exact location and provides information about the nearby attractions or places of interest.

Projection-based AR uses projected light by combining cameras to 3D sensing systems (e.g. depth cameras) and allowing digital displays to appear on any surface. This type of AR has applications in operations, manufacturing, and other fields. A popular example may be represented by the projected images on buildings, usually at Christmas markets at night.

The last type of AR is the superimposition-based AR. This one replaces partially or wholly an existing object from the real world with an augmented view of it. A famous example of this technology being used is an application from IKEA, with which customers can superimpose furniture from the online retailer right in their homes, to see how it would look like. Image recognition plays a vital role here, otherwise there is nothing to be replaced.

AR has uses in entertainment, education, art, tourism, and cultural heritage, as well as others. Today, the number of smartphone users is around 3.5 billion [39], making Mobile AR (MAR) more accessible than ever via social media apps, gaming and others.

Cultural Heritage

According to Bill Ivey, *Cultural Heritage (CH)* "tells us where we came from by preserving and presenting voices from the past, grounding us in the linkages of family, community, ethnicity, and nationality, giving us our creative vocabulary" [35]. Cultural heritage can also be split in tangible and intangible. Tangible CH refers to physical artifacts, including artistic creations and other tangible products that have a cultural importance. Intangible CH refers to non-physical practices, expressions, knowledge, artifacts and cultural spaces associated with communities, groups and individuals [2]. Due to the paramount importance of heritage, cultural institutions are safeguarding the world's art, culture, history and heritage for hundreds of years already. With the new technologies of today and tomorrow, this safeguarding mission may become easier, with an even wider audience – a worldwide one. When the access to the art and heritage of people's past is restricted, understanding human nature becomes more difficult. Digital technologies and web-based communication platforms remove the obstacles in disseminating knowledge and cultural heritage to people.

Being already an advanced technology, AR is a great way to meet the needs of cultural heritage. Using AR in the cultural heritage context maximizes visitor's satisfaction and offers a unique, personalized experience to each tourist [29]. If the user experience and the usability of the application are also taken into consideration, it is even more accessible for the cultural heritage to reach its goal of being noticed.

In this paper the taxonomy for *culture* is empirically linked to cultural heritage, tourism, art, and museums, based on the surveyed papers and on the lack of a standard taxonomy for this domain.

RELATED WORK

As AR started to develop in the cultural heritage field, there are already literature reviews which aim to present the trends in this domain.

Aliprantis et al. [3] surveyed in 2019 the trends in building AR applications for the cultural heritage field. They divided the different AR approaches into 8 categories: serious games, personalization, AR reconstruction, projection display, Semantics/Linked Open Data, cultural UX evaluation, Context Awareness and digital storytelling. Considering the amount of the various approaches, the authors assume that the evaluation of the Cultural UX is one of the most active fields in AR applications for cultural heritage, as institutions constantly try to adapt to their visitor's needs. Most of the papers analyzed in this survey used the simplest evaluation methods, as a short questionnaire, in order to obtain feedback from the visitors. Another active field in AR applications for cultural heritage is digital reconstruction. Recent works in the cultural heritage field present multiple and different techniques to virtually reconstruct cultural relics or monuments and provide an immersive experience to their users.

Bekele et al. [4] surveyed augmented, virtual, and mixed reality from a cultural heritage perspective. On the AR topic, the authors identified that AR applications in the cultural heritage domain frequently use marked-based, markerless, and hybrid tracking approaches. In a hybrid approach, the camera of the device and electromagnetic, inertial, and acoustic sensors are used. They categorize AR systems in two types: indoor and outdoor AR. Indoor AR uses marker-based tracking, and usually does not require the use of GPS. Outdoor AR relies heavily on markerless and hybrid tracking. They also classified the purpose of augmented, virtual and mixed reality when used for cultural heritage:

- Education aims at enabling users to learn the historical aspects.
- Exhibition enhancement at physical museums and heritage sites.
- Exploration supports users in visualizing and exploring historical and current views of cultural heritage.
- Reconstruction enables users to picture and interact with reconstructed historical views of cultural heritage.
- Virtual museums simulate and present cultural heritage (both tangible and intangible) to the public.

The survey showed that AR is preferable for exhibition enhancement.

Another survey presented the cultural heritage in markerless AR [21]. The authors classify the AR techniques in two main categories: vision-based AR and location-based AR. The study summarizes the existing research on markerless AR, both for indoor and outdoor use. It also identifies four main issues related to AR when used for cultural heritage. The first concern is related to registration and it refers to the compatibility of augmented objects with the real environment. The second topic is reconstruction, which is a construction of virtual objects in a way to replicate the original building – this is usually done by using 3D scanning techniques. The third issue refers to tracking, where a high level of accuracy and a low level of latency are the key requirements. The last challenge for markerless AR is the location, which needs to be set up correctly.

None of the presented surveys are systematic, with a research methodology hard to reproduce.

METHODOLOGY

In this study, a Systematic Literature Review was conducted on usability, mobile AR and cultural heritage. Being systematic, it aims to set up a search protocol, identify all studies that would meet the eligibility criteria and present the findings of the included studies [24]. The study aims to answer the following research questions:

- What kind of usability evaluation did researchers of cultural heritage based mobile augmented reality applications perform?
- How do these types of applications perform? What were the outcomes of the usability evaluation?

The interrogated databases were Web of Science, Scopus, IEEE and Springer. The search was based on the logical expression "mobile augmented reality" AND "cultural heritage" AND ("user experience" OR "usability"). In all four databases, the search was limited to papers published between 2010 and 2020.

In total, 88 papers were found by searching with the above-mentioned keywords. In order to be eligible to be included in this survey, a study needed to meet the following criteria:

- The field of study or the application mentioned in the paper to have as a platform a mobile device (either smartphone or tablet).
- The published language of the study to be English.
- At least one usability or user experience finding regarding MAR used for cultural heritage.
- The study area to be related to culture (here, we took into consideration all of cultural heritage, art, museums, historical sites, and tourism).

Out of the 88 found papers, 57 were excluded due to not meeting one or more eligibility criteria described above, and 31 were kept for further analyzing. Out of the kept researches, 18 of them were found on Springer, 16 on Scopus, 13 on Web of Science and only 2 in IEEE. Some of the studies were found in more than one database.

RESULTS

All the 31 kept papers have some factors in common, such as the applied domain is cultural heritage, refers to a mobile AR application and has a usability study.

In Table 1 a full comparison of the AR environment and AR types from the surveyed papers is presented. For the environment category, the applicability of the mobile application was considered, as it was created to be used indoors or outdoors. The other category refers to the AR type used – here, the split is made between location-based (when GPS and different sensors are used) and image-based (when QR codes or 2D images are used). Out of the 31 studies, 3 of them [10,27,33] considered indoors use and the AR was based on images. In these cases, the user points the mobile device towards a marker placed in a physical scene and an associated image or painting is displayed on the smartphone's screen. Another study [37] considered indoors use based on location. In this case, a 3D model of a fortified church was virtually displayed on the screen after the physical place was scanned by the sensors first. Another study [13] offer application options for both indoor and outdoor environments based on GPS and sensor tracking. [31] presents different case-studies which employ both indoor and outdoor environments, as well as AR based on location and on images. [7] detects the user's position by using the compass and the accelerometer and when the Point of Interest (POI) is closer than 5 meters, the system uses image-based mode.

Table 1. A full comparison of AR environment and AR types in the surveyed papers

Study	Environment (Indoor/Outdoor)	Based on location (GPS, sensor)/Based on images (QR/2D)
[10,27,33]	Indoor	Based on images
[37]	Indoor	Based on location
[13]	Both	Based on location
[7,31]	Both	Both
[1,5,6,8,9,11,12,14–20,22,23,25,26,28,30,32,34,36,38]	Outdoor	Based on location

The other 24 studies relate to mobile applications for outdoors, and all of them employ the location-based AR, which uses GPS and sensors. What can be noticed here is that outdoor applications are prevalent in cultural heritage, and they all are markerless AR, based on location. A reason for that might be, as stated in [13], that enjoyment is higher in the outdoor settings due to the high level of relevance and realism.

Table 2. Usability testing methods empoloyed in the surveyed papers

Application	Questionnaire	Focus Group	User Testing	Interview
3DGuides platform [22], MobiAR [25]	x			x
ARAC Maps [10], 3 interfaces (map, list AR) [12], ArcHIVE 4Any [19], Flaneur [15], Changdeokgung Palace [20]	x		x	
Carpano [33], Mobile AR guide for tourists [34], TowerAR [9], Virtual/Augmented Gallery [27], Transit Assistant [18], Finnish Outdoor museum [32], Ai Guang Zhan [8], SitCity (proposed framework) [36], MixAR [1], "Deoksugung, in my hands" and DublinAR [16,23], MAR app for Melaka Heritage Sites [38], OvidAR [6], MAR app to revive a demolished Reformed Church in Brașov [5], The Historical Tour Guide [14], 3D model of fortified church [37]	x			
MAR application geo-located and gamified [28], Archaeological park [30], 2 apps: MTL Urban Museum and MetaGuide [13], AR workshops [31]			x	
Open City Museum [17]			x	x
Seraj [7], Millennia Road [26]	x		x	x
KnossosAR [11]	x	x		x

In terms of usability, four major testing methods were identified: questionnaires, focus groups, user testing and interviews. Questionnaires were mostly used after the users interacted with the application on-site and frequently included questions targeting the ease of use, the usefulness of the application or the perceived enjoyment. Another method of usability testing which was used only in one study [11] is the focus group. The participants tested the KnossosAR application in four focus groups of four participants each in order to facilitate their live observation by the app developers. Besides the focus group, a questionnaire was also needed to express the overall quality of experience and document any remarks. And finally, a semi-structured interview followed the questionnaire, to offer participants the opportunity to clarify any concerns and suggest additional enhancements. The third usability testing method is represented by user testing. This implies the evaluation of the proposed application in terms of provided functionality and design ease. In user testing, the participants have to perform a set of tasks by using the application and usually the time taken to perform them is measured. In [13], a portable eye-tracking device was also used, as well as audio recordings of verbal interactions between the user and the guide. The last testing method is the interview, implying a face-to-face discussion, either completely informal or semi-structured between the users and the observers. In all surveyed articles, the interview was used as a secondary testing method.

In Table 2 a comparison of the usability testing methods is provided. At first glance, it can be noticed that the questionnaire alone was the most used method in the surveyed papers. In most of the studies, the users stated that the application is easy to use and intuitive. They also appreciate the clarity of the interface, the shape and colors of the graphic elements. Several papers [5,6,11,13,14,16] confirmed the enjoyment the users have while using the mobile AR application. Some of those [5,6,11,26,38] also obtained great results regarding the perceived usefulness of the application. [14] even showed that both perceived usefulness and perceived enjoyment had a strong positive effect on the intention of using this type of application. Here, the ease of use can also be a good predictor on the intention to use the app. However, as noted by the authors, tourists have a higher intention in using a mobile AR app for cultural heritage than local residents, who do not feel the same need in their hometown. Likewise, people want to use this type of applications because they enjoy the experience, but also because it helps them achieve some learning objective. Some works as [26], [7] and [11] used three usability testing methods, making their results even more reliable. In all three studies the perceived usefulness, ease of use and enjoyment were confirmed.

In certain studies the usability testing results mention in addition the impact on learning. [34] correlated the ease of use with the ease of learning and [31] suggest that mobile AR could be effective in learning processes as a complement to conventional training. In [38], 94% out of 200 participants preferred the mobile AR application compared to traditional methods of learning. Also, a mobile AR app may increase the users' interest in learning about cultural heritage, as it happened in [37].

A few of the usability downsizes included the high battery consumption for cases in which the AR camera and the GPS were working simultaneously [28]. In [1], the lack of engagement when using the mobile AR app was linked to the low screen resolutions that were set up to ensure a fair exchange between the experience fluidity and the available computational results. Also, the display in mobiles is prone to outdoors light reflections which may cause additional efforts for the visitors. In another study [26], 11 out of 30 participants felt that keeping their heads down to watch the smartphone disconnects them and prevents them from enjoying the surroundings, and five of them found it was "too heavy" to hold the smartphone up while interacting with the guide. In one more study [10] the users said the menu handle was too small – a minor usability flaw which was corrected after the feedback session. Another missing feature was an interactive tutorial when first starting the app [7,34]. Research in [11] showed that users preferred images, audio and narration to textual information, as they claimed it distracted their attention when looking at the POI itself. Furthermore, map environments overcrowded with POIs are restricting the usability [18], as the users are often required to tap on several markers only to be able to locate a particular POI. In the KnossosAR application [11], the authors addressed the occlusion challenge which is usually met in location-based AR applications. They created an efficient method for estimating the field of view of the user, so the POIs would not be obstructed by physical obstacles anymore.

As per gender differences in usability for such mobile applications, [8] found no significant disparities. Nevertheless, [34] found that females rated the experience with AR more satisfying and the interaction better and more intuitive than males.

In [27], 24 out of the 25 study participants expressed their desire to see this type of technology adopted by museums.

DISCUSSION

The intention of this study was to survey the research in usability testing of mobile AR applications in the cultural heritage field.

There are some limitations to the study that need to be considered. Firstly, only four databases were interrogated, with a specific set of keywords. Future researches could include other relevant databases, such as ACM. Using different keywords might have yielded different results. Considering this, there may be research papers on the subject which are not indexed in the searched databases. Secondly, as a taxonomy for cultural heritage was not found, studies from related fields were included empirically (e.g. tourism, art etc.).

The study hopes to shed a light on the overall user experience of mobile AR applications for cultural heritage. AR has become a great way in helping to preserve, document and explore cultural heritage by bringing people pieces of the past in an interactive and engaging manner. Most of the surveyed papers addressed the usability of their applications via questionnaires. However, only a few studies employed more usability testing methods, especially those which are strictly targeting the interface layout (e.g. eye-tracking devices). Moreover, nearly all the papers referred to outdoor use and location-based AR, as outdoors AR may offer a more natural experience, and being location-based, it offers more accurate results, as sensors and GPS do not rely on lighting conditions.

CONCLUSIONS

This paper evaluated the research in usability testing of mobile AR applications for cultural heritage. With the aim to be a systematic literature review study, four databases (IEEE, Scopus, Web of Science and Springer) were interrogated by using specific logical expressions. The search found 88 papers, out of which only 31 met the eligibility criteria described in Methodology. All maintained papers have some elements in common, such as the applied domain which is cultural heritage, and they also refer to a mobile AR application and have a usability study.

Findings revealed that 86% of the papers (without considering the 3 studies which used both types of AR [7,13,31]) referred to outdoor use and location-based AR, as outdoors AR may offer a more natural experience, and being location-based, it offers more accurate results. Only a few studies (14%) targeted the indoors use and were based solely on 2D image recognition.

Moreover, the results showed that most of the applications created were easy to use, intuitive and the study participants enjoyed using them. Also, users appreciate a clear and simple interface. The ease to use the application has a strong impact on the intention to use it. In the same way, the easier to use the mobile AR application is, the easier it is for the user to learn about the content presented. In one study, females rated the experience with AR more satisfying and the interaction better and more intuitive than males. What users did not particularly like were the missing tutorials when first starting the app, the graphic elements too small to interact with, reading a lot of content on a small screen or making efforts to check the screen due to the high brightness conditions of the outdoors, or even having to hold the smartphone up for too long.

What seems to be appreciated in mobile AR apps for cultural heritage are the tutorials from the beginning, the arrows showing how to use the app, the easy to find POIs and a simple, yet intuitive interface.

The cultural heritage domain can benefit tremendously from the mobile AR applications, as half the population owns a smartphone and it is a great way of preserving, documenting, and exploring all the values it holds. The present survey showed that there is still room for improvement in regard to the user experience of mobile AR applications for cultural heritage. Nevertheless, the field has become quite popular in the past few years, promising an even bigger popularity change.

REFERENCES

[1] Telmo Adao, Luis Padua, David Narciso, Joaquim Joao Sousa, Luis Agrellos, Emanuel Peres, and Luis Magalhaes. 2019. MixAR: A Multi-Tracking Mixed Reality System to Visualize Virtual Ancient Buildings Aligned Upon Ruins. *J. Inf. Technol. Res.* 12, 4 (December 2019), 1–33. DOI:https://doi.org/10.4018/JITR.2019100101

[2] Yahaya Ahmad. 2006. *The Scope and Definitions of Heritage: From Tangible to Intangible*. International Journal of Heritage Studies.

[3] John Aliprantis and George Caridakis. 2019. A Survey of Augmented Reality Applications in Cultural Heritage: *International Journal of Computational Methods in Heritage Science* 3, 2 (July 2019), 118–147. DOI:https://doi.org/10.4018/IJCMHS.2019070107

[4] Mafkereseb Kassahun Bekele, Roberto Pierdicca, Emanuele Frontoni, Eva Savina Malinverni, and James Gain. 2018. A Survey of Augmented, Virtual, and Mixed Reality for Cultural Heritage. *J. Comput. Cult. Herit.* 11, 2 (March 2018), 7:1–7:36. DOI:https://doi.org/10.1145/3145534

[5] Razvan Gabriel Boboc, Florin Girbacia, Mihai Duguleana, and Ales Tavcar. 2017. *A handheld Augmented Reality to revive a demolished Reformed Church from Brasov*. Assoc Computing Machinery, New York. DOI:https://doi.org/10.1145/3110292.3110311

[6] Rizvan Gabriel Boboc, Mihai Duguleana, Gheorghe-Daniel Voinea, Cristian-Cezar Postelnicu, Dorin-Mircea Popovici, and Marcello Carrozzino. 2019. Mobile Augmented Reality for Cultural Heritage: Following the Footsteps of Ovid among Different Locations in Europe. *Sustainability* 11, 4 (February 2019), 1167. DOI:https://doi.org/10.3390/su11041167

[7] Fatiha Bousbahi and Bayan Boreggah. 2018. *Mobile Augmented Reality Adaptation through Smartphone Device Based Hybrid Tracking to Support Cultural Heritage Experience*. Assoc Computing Machinery, New York. DOI:https://doi.org/10.1145/3289100.3289109

[8] Chen-Chiou Chiu and Lai-Chung Lee. 2018. System satisfaction survey for the App to integrate search and augmented reality with geographical information technology. *Microsyst Technol* 24, 1 (January 2018), 319–341. DOI:https://doi.org/10.1007/s00542-017-3333-9

[9] M. Duguleana, R. Brodi, F. Girbacia, C. Postelnicu, O. Machidon, and M. Carrozzino. 2016. Time-travelling with mobile augmented reality: A case study on the piazza dei Miracoli. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10058 LNCS, (2016), 902–912. DOI:https://doi.org/10.1007/978-3-319-48496-9_73

[10] Daniel Eggert, Dennis Hücker, and Volker Paelke. 2014. Augmented Reality Visualization of Archeological Data. In *Cartography from Pole to Pole: Selected Contributions to the XXVIth International Conference of the ICA, Dresden 2013*, Manfred Buchroithner, Nikolas Prechtel and Dirk Burghardt (eds.). Springer, Berlin, Heidelberg, 203–216. DOI:https://doi.org/10.1007/978-3-642-32618-9_15

[11] P. Galatis, D. Gavalas, V. Kasapakis, G. Pantziou, and C. Zaroliagis. 2016. Mobile augmented reality guides in cultural heritage. DOI:https://doi.org/10.4108/eai.30-11-2016.2266954

[12] Dion Hoe-Lian Goh, Chei Sian Lee, and Khasfariyati Razikin. 2011. Comparative Evaluation of Interfaces for Presenting Location-Based Information on Mobile Devices. In *Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation* (Lecture Notes in Computer Science), Springer, Berlin, Heidelberg, 237–246. DOI:https://doi.org/10.1007/978-3-642-24826-9_30

[13] Jason M. Harley, Eric G. Poitras, Amanda Jarrell, Melissa C. Duffy, and Susanne P. Lajoie. 2016. Comparing virtual and location-based augmented reality mobile learning: emotions and learning outcomes. *Education Tech Research Dev* 64, 3 (June 2016), 359–388. DOI:https://doi.org/10.1007/s11423-015-9420-7

[14] Anne-Cecilie Haugstvedt and John Krogstie. 2012. Mobile Augmented Reality for Cultural Heritage: A Technology Acceptance Study. In *2012 Ieee International Symposium on Mixed and Augmented Reality (ismar) - Science and Technology*. Ieee, New York, 247–255.

[15] Anastasia Ioannidi, Damianos Gavalas, and Vlasios Kasapakis. 2017. Flaneur: Augmented exploration of the architectural urbanscape. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, Heraklion, Greece, 529–533. DOI:https://doi.org/10.1109/ISCC.2017.8024582

[16] Timothy Hyungsoo Jung, Hyunae Lee, Namho Chung, and M. Claudia Tom Dieck. 2018. Cross-cultural differences in adopting mobile augmented reality at cultural heritage tourism sites. *Int. J. Contemp. Hosp.*

Manag. 30, 3 (2018), 1621–1645. DOI:https://doi.org/10.1108/IJCHM-02-2017-0084

[17] Georgios Kallergis, Marios Christoulakis, Aimilios Diakakis, Marios Ioannidis, Iasonas Paterakis, Nefeli Manoudaki, Marianthi Liapi, and Konstantinos-Alketas Oungrinis. 2020. Open City Museum: Unveiling the Cultural Heritage of Athens Through an -Augmented Reality Based- Time Leap. In *Culture and Computing* (Lecture Notes in Computer Science), Springer International Publishing, Cham, 156–171. DOI:https://doi.org/10.1007/978-3-030-50267-6_13

[18] Manousos Kamilakis, Damianos Gavalas, and Christos Zaroliagis. 2016. Mobile User Experience in Augmented Reality vs. Maps Interfaces: A Case Study in Public Transportation. In *Augmented Reality, Virtual Reality, and Computer Graphics* (Lecture Notes in Computer Science), Springer International Publishing, Cham, 388–396. DOI:https://doi.org/10.1007/978-3-319-40621-3_27

[19] Manolya Kavakli. 2015. A people-centric framework for mobile augmented reality systems (MARS) design: ArcHIVE 4Any. *Human-centric Comput. Inf. Sci.* 5, (December 2015), 37. DOI:https://doi.org/10.1186/s13673-015-0055-9

[20] Hayun Kim, Tamas Matuszka, Jea-In Kim, Jungwha Kim, and Woontack Woo. 2017. Ontology-based mobile augmented reality in cultural heritage sites: information modeling and user study. *Multimed. Tools Appl.* 76, 24 (December 2017), 26001–26029. DOI:https://doi.org/10.1007/s11042-017-4868-6

[21] H. Kolivand, A. El Rhalibi, S. Abdulazeez, and P. Praiwattana. 2018. Cultural Heritage in Marker-Less Augmented Reality: A Survey. In *IntechOpen*, D. Turcanu-Carutiu and R.-M. Ion (eds.). Intechopen, London. DOI:https://doi.org/10.5772/intechopen.80975

[22] K.I. Kotsopoulos, P. Chourdaki, D. Tsolis, R. Antoniadis, G. Pavlidis, and N. Assimakopoulos. 2019. An authoring platform for developing smart apps which elevate cultural heritage experiences: A system dynamics approach in gamification. *Journal of Ambient Intelligence and Humanized Computing* (2019). DOI:https://doi.org/10.1007/s12652-019-01505-w

[23] Hyunae Lee, Namho Chung, and Timothy Jung. 2015. Examining the Cultural Differences in Acceptance of Mobile Augmented Reality: Comparison of South Korea and Ireland. In *Information and Communication Technologies in Tourism 2015*, Springer International Publishing, Cham, 477–491. DOI:https://doi.org/10.1007/978-3-319-14343-9_35

[24] Alessandro Liberati, Douglas G. Altman, Jennifer Tetzlaff, Cynthia Mulrow, Peter C. Gøtzsche, John P. A. Ioannidis, Mike Clarke, P. J. Devereaux, Jos Kleijnen, and David Moher. 2009. The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration. *PLoS Med* 6, 7 (July 2009). DOI:https://doi.org/10.1371/journal.pmed.1000100

[25] María Teresa Linaza, David Marimón, Paula Carrasco, Roberto Álvarez, Javier Montesa, Salvador Ramón Aguilar, and Gorka Diez. 2012. Evaluation of Mobile Augmented Reality Applications for Tourism Destinations. In *Information and Communication Technologies in Tourism 2012*, Springer, Vienna, 260–271. DOI:https://doi.org/10.1007/978-3-7091-1142-0_23

[26] E. Litvak and T. Kuflik. 2020. Enhancing cultural heritage outdoor experience with augmented-reality smart glasses. *Personal and Ubiquitous Computing* (2020). DOI:https://doi.org/10.1007/s00779-020-01366-7

[27] Michele Mallia, Marcello Carrozzino, Chiara Evangelista, and Massimo Bergamasco. 2019. Automatic Creation of a Virtual/Augmented Gallery Based on User Defined Queries on Online Public Repositories. In *VR Technologies in Cultural Heritage* (Communications in Computer and Information Science), Springer International Publishing, Cham, 135–147. DOI:https://doi.org/10.1007/978-3-030-05819-7_11

[28] C. Panou, L. Ragia, D. Dimelli, and K. Mania. 2018. Outdoors mobile augmented reality application visualizing 3D reconstructed historical monuments. 59–67.

[29] C. Perra, E. Grigoriou, A. Liotta, W. Song, C. Usai, and D. Giusto. 2019. Augmented reality for cultural heritage education. 333–336. DOI:https://doi.org/10.1109/ICCE-Berlin47944.2019.8966211

[30] Roberto Pierdicca, Emanuele Frontoni, Primo Zingaretti, Eva Savina Malinverni, Andrea Galli, Ernesto Marcheggiani, and Carlos Smaniotto Costa. 2016. Cyberarchaeology: Improved Way Findings for Archaeological Parks Through Mobile Augmented Reality. In *Augmented Reality, Virtual Reality, and Computer Graphics, Pt Ii*, L. T. DePaolis and A. Mongelli (eds.). Springer International Publishing Ag, Cham, 172–185. DOI:https://doi.org/10.1007/978-3-319-40651-0_14

[31] Ernest Redondo Domínguez, David Fonseca Escudero, Albert Sánchez Riera, and Isidro Navarro Delgado. 2014. Mobile learning in the field of Architecture and Building Construction. A case study analysis. *Int J Educ Technol High Educ* 11, 1 (January 2014), 152–174. DOI:https://doi.org/10.7238/rusc.v11i1.1844

[32] Kaapo Seppälä, Olli I. Heimo, Timo Korkalainen, Juho Pääkylä, Jussi Latvala, Seppo Helle, Lauri Härkänen, Sami Jokela, Lauri Järvenpää, Frans Saukko, Lauri

Viinikkala, Tuomas Mäkilä, and Teijo Lehtonen. 2016. Examining User Experience in an Augmented Reality Adventure Game: Case Luostarinmäki Handicrafts Museum. In *Technology and Intimacy: Choice or Coercion* (IFIP Advances in Information and Communication Technology), Springer International Publishing, Cham, 257–276. DOI:https://doi.org/10.1007/978-3-319-44805-3_21

[33] S. Spacca, E. Dellapiana, and A. Sanna. 2018. Promoting industrial cultural heritage by augmented reality: Application and assessment. *Open Cybernetics and Systemics Journal* 12, 1 (2018), 61–71. DOI:https://doi.org/10.2174/1874110X01812010061

[34] D. Stoelák, F. Škola, and F. Liarokapis. 2016. Examining user experiences in a mobile augmented reality tourist guide. DOI:https://doi.org/10.1145/2910674.2935835

[35] Ann Marie Sullivan. Cultural Heritage & New Media: A Future for the Past, 15 J. Marshall Rev. Intell. Prop. L. 604 (2016). *NEW MEDIA*, 44.

[36] Vanessa Camilleri. 2020. Augmented Reality in Cultural Heritage: Designing for Mobile AR User Experiences. In *Rediscovering Heritage Through Technology: A Collection of Innovative Research Case Studies That Are Reworking The Way We Experience Heritage*, Dylan Seychell and Alexiei Dingli (eds.). Springer International Publishing, Cham, 215–237. DOI:https://doi.org/10.1007/978-3-030-36107-5_11

[37] Gheorghe-Daniel Voinea, Florin Girbacia, Cristian Cezar Postelnicu, and Anabela Marto. 2019. Exploring Cultural Heritage Using Augmented Reality Through Google's Project Tango and ARCore. In *Vr Technologies in Cultural Heritage*, M. Duguleana, M. Carrozzino, M. Gams and I. Tanea (eds.). Springer International Publishing Ag, Cham, 93–106. DOI:https://doi.org/10.1007/978-3-030-05819-7_8

[38] Syamsul Bahrin Zaibon, Ulka Chandini Pendit, and Juliana Aida Abu Bakar. 2015. Applicability of Mobile Augmented Reality Usage at Melaka Cultural Heritage Sites. In *Proceedings of the 5th International Conference on Computing & Informatics*, Z. Jamaludin, N. ChePa, W. H. W. Ishak and S. B. Zaibon (eds.). Univ Utari Malaysia-Uum, Sintok, 235–240.

[39] Smartphone users worldwide 2020. *Statista*. Retrieved July 21, 2020 from https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

Interactive Assembly Simulation in an Immersive Virtual Environment

Cătălin Moldovan
Technical University of Cluj-Napoca
Str. G. Barițiu 28, 400027,
Cluj-Napoca, România
catalin.moldovan97@gmail.com

Adrian Sabou
Technical University of Cluj-Napoca
Str. G. Barițiu 28, 400027,
Cluj-Napoca, România
adrian.sabou@cs.utcluj.ro

DOI: 10.37789/rochi.2020.1.1.22

ABSTRACT

This paper describes the development of a Virtual Reality (VR) simulation application for educational purposes in assembly production. People are able to retain more information by simulating real experiences. Nowadays, modern technology can be used to achieve better results in a virtual reality learning environment, being available to everyone. It is predicted to be more common and low-priced in the future. Interactive user gestures and immersive VR technologies are used to develop remote solutions for engineering students by matching the product components in the proper order and location. This research provides instructional assembly methods and natural experience in interacting with elements, both in a dynamic space or in a sitting position.

Author Keywords

Virtual Reality; Leap Motion; Assembly Simulation; OpenGL; Educational Application;

ACM Classification Keywords

H.5.m. Information interfaces and presentation: Human-Computer Interaction; Interaction Techniques; Gestures;

General Terms

Virtual Reality; Computer graphics; Algorithms; Input devices; Head-mounted display;

INTRODUCTION

Modern graphic techniques have changed the way people perceive the virtual world, developing a new approach of understanding problem solving. Many devices and computer software allow us to convert human interaction into tridimensional (3D) data, in order to replicate the human hand model on screen. So much time is being waste to learn a new user interface of an application for a novice computer user. People can adapt faster in an immersive experience, than using a desktop application. The experience is achieved from the first movements and gestures performed by the user, without a tutorial or guided texts. The virtual contact with the environment is based on user's natural interaction intuition.

Virtual reality users are growing from day to day, from 200.000 users in 2014, to over 171.000.000 users in 2018 [1]. It has become widely spread in the game industry and now is expanding on manufacturing and medical industry. The first Oculus Rift prototype was released in 2012 by Palmer Luckey and the game engine designer of Doom franchise, John Carmack. This start was noticed by many big companies and quickly made their way into business industry, becoming available to the public.

Nowadays many educational facilities, like mechatronics, machine building, aviation or construction technical college are missing essential equipment to train students. In many cases, students know the theoretical part and even the main process, but this data is forgotten in time due to lack of real demonstration or simulation. The objectives of this paper is to increase learning performance by using modern technology to simulate real-life assembly. The graphics environment is designed on a long-term human retention, based on visual and interactive processes. Simple structures can create an attractive and innovative space scene, so that users can better perceive the objects' depths and distances, based on 3D placement and shadows [2].

The main keys are accuracy and user comfort. Having these in mind, environment accommodation during the virtual simulation will no longer create confusion or motion sickness, while wearing a headset. The graphic scene rendered on the screen is designed to send the mind into a more immersive 3D experience, having full control on the virtual space.

After a virtual experience session, students are more trained to apply what they learned in real situation. Within a virtual assembly session, people might try to explore wrong alternatives, deviating from the base assembly process, which might lead to bad consequences, like breakage, nonfunctional devices, short circuit or even accidents. We cannot experience wrong possibilities in real life, but in a simulation, everything is possible, leading the user to identify the correct assembly process and to understand the incompatibility of certain parts.

The proposed solution makes use of the head movements and natural hand interactions of user's actions, in order to simulate a virtual model assembly, using tracking devices like sensors and cameras. This paper elaborates on models orientation, which tries to simulate object manipulation by human in real world and presents a fresh new graphic engine, as the project's foundation. The VR application was built using OpenGL library with the Oculus Rift Development Kit 2 (DK2) headset and LeapMotion controller for hand motion tracking. The main focus is aimed on simple and complex techniques of interaction, in order to reach the behavior of a real life object manipulation.

The rest of the paper is structured as follows. In the next section, we present similar ideas and projects designed by students and researchers. In the following four sections we describe the major graphics engine infrastructure components, the interaction algorithms developed for object manipulation, parent-child relationship between two objects and an overview of the 3D model creation. In the next two sections we discuss about different guided methods of model assembly and the performance of level completion. In the last section we describe the experience gained from implementing the overall assembly project, we present our conclusions and future project improvements.

RELATED WORKS

Zhao et al. [3] present a VR simulation game for manufacturing education by interacting with LEGO pieces, using wireless controller in hand. The overall goals of the project is to provide engineering students with a set of scenarios to practice their skills at craft production. They describe the development of the immersive experience by using a custom fitted headset with Tobii eye-tracking technology. The environment is based on assembly station for the users to go through and accomplish a set of requirements. The user has to choose the components in order to start crafting the production process.

Pujol-Tost and Phil [4] analyze the influence of computational VR interactivity in the learning process, based on response speed, range of things that can be changed and naturality of communication. They analyze them all by showing how they involve different learning and interaction strategies. The source of motivation in the learning process has proven to be higher, the more interactive and immersive the experience is. The main key point is the equality of conditions when user interacts with the content, simulating similar real experience. As formal educational environments have demonstrated a positive attitude towards interactive devices, they continue to evolve and be more accessible to people all over the world.

Zimmons and Panter [5] proceed an experiment of college-age participants on how visual elements like lighting, surface detail and task performance influence the sense of presence of participants in a virtual environment. Based on some graphics conditions, the experiment uses a head mounted display and a joystick, with a trigger function to grab objects from scene. The study suggested that rendering quality environments is not significantly affecting the perception of depth or user's precision. A major difference of spatial orientation was determined not to be equal between man and women.

Pop and Sabou [6] use the LeapMotion controller to interact with virtual scene, using Unity Game Engine. They present an approach to dynamic data visualization and manipulation through a server-side application, based on hand gestures and head movement and orientation, tracked from phone's gyroscopic information.

Galais et al. [7] evaluated gestural interaction using LeapMotion and a traditional interaction device, using gamepad controllers. The comparative study is based on the cognitive load and performance of object manipulation, performed by 11 experienced users and 8 novice users. The results indicate a higher execution time and users' errors during gestural interaction with the LeapMotion device rather than using a controller. The main limitations are intermittent hand tracking and the difficulty in interacting and reaching the object as no haptic feedback is provided.

Boud et al. [8] conducted a series of experiments to compare assembly completion times after participants study an engineering drawing or an assembly plan, using VR and Augmented Reality (AR) as training media. In order to achieve simple goals of interacting with objects, like reaching an object, grasping or placing objects, which require different levels of haptic and visual guidance. A VR manufacturing environment allows users to manipulate objects without the use of the real objects and also to be trained for an assembly operation during a product's design cycle, before an actual physical prototype has been manufactured. The participants suggested that immersive VR was more intuitive as they were able to manipulate 3D objects in a 3D space. AR can therefore facilitate fast learning for simple assembly tasks, as it allows the user to have tactile feedback through the manipulation of the real objects.

Baggett and Ehrenfeucht [9] present how to design instructions that show and describe a step by step procedure using a hierarchical structure. The structure of an object can be represented by a labelled tree, as each node has a value, which presents the object's name. The tree shows the model breakdown into subassemblies and subsubassemblies, the procedure description, which tells the actions performed and the goal to build the complete model, which can be divided in subgoals. The paper tests the performance in assembly from memory, as the object is correctly built by the user. The best performance is achieved when combining a top-down approach with a sequential execution of actions. It is also demonstrated that the presentation of instructions via a video can improve performance of assembly operations. Humans have a remarkable ability to store visual information over short periods of time. Simply seeing the assemblies being built was sufficient for experienced participants to be able to develop assembly plans.

GRAPHICS ENGINE

Creating a lightweight graphic engine for this project, focused on render algorithms and interaction methods, might be useful for the freedom of using minimum computer resources. The free real-time 3D creation platforms Unity and Unreal Engine 4 offer a user interface, many options and properties for use to design and conceptualise the virtual world. Figure 1 displays the engine specifications of different free engines. Code files, models and materials are efficient organized for the user and the real time application scene makes designing easier and faster. Visual scripting technique lets user create scene content events without coding skills. Both engines have many plugins and scene creation tools available on their asset store. The complexity as well as the numerous integrated features contribute to the final application size, providing additional specifications which are not always needed.

Our application is based on free C++ libraries for graphic software developing, based on OpenGL Shading Language (GLSL). The efficiency in using a fresh new engine, is based on extensions, quality optimization and memory allocation. As follows, there are also disadvantages of building a custom engine as speed processing, data partitioning, threads execution model or the number of features. The application is designed as a flexible tool based on virtual interaction structure organized in a software architecture, having the possibility to study system response to external hardware, resource management and 3D transformation concepts.

Godot Engine is a free and open-source game engine which at first sight, it would be the best choice of developing a small application, aimed on 2D and simple 3D games. The issue might be more of scaling, which might affect the performance, but overall it is not at the level of support, features and functionality compared to other engines. It has its own programming language GDScript, but similar to our solution, the application's configuration has to be made manually by the user [10].

The overall engine solution comes with visual effects for lightning, shadow mapping, environment mapping, reflective materials properties creation, text and video rendering, Table 1. Sounds and animation elements were used for focusing user attention on the action location. The main limitations identified are mainly focused on the scene realism, low on extensions and complex application structure.

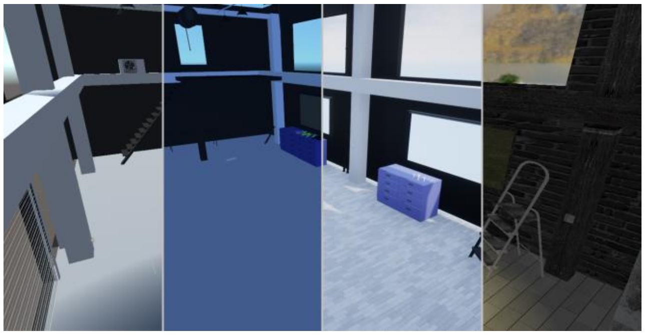


Figure 1. Graphic Engines scene comparison, from left to right: Unity, Godot, Unreal Engine, our engine solution

Engine Service	Proposed Engine	Unity	UE4	Godot 3.0
Programming language	C	C#	C++	GDScript C/C++ C#
Framework	OpenGL	Direct3D OpenGL Vulkan	Direct3D	OpenGL
Dimensions	3D	2D, 3D	2D, 3D	2D, 3D
Storage Space	130 MB	4 - 7 GB	10-15 GB	500 MB
VR support	Yes	Yes	Yes	Yes

Table 1. Engines specifications

The immersive components

The overall VR session is based on the communication between human and hardware components. The immersive environment is achieved by synchronizing the hand interaction, head position and orientation with the virtual world, having at least 60 frames per seconds displayed on the headset's screen. Communication between user and system is done through input devices, by sending the human movement and interaction information to the computer and output devices, which receive the processed data back to the user. The human head is traced by the camera-based system, which uses filters to capture infrared light trackers on the back of the Oculus headset case. LeapMotion sensors and the monochromatic camera allow the user to interact within the virtual scene. The software is processing each human hand bone, tracked in the device's range and store them as data, which can be accessed by an API for each available frame processed. The information is used to trigger scene events, recognize hand gestures and render the skeleton of the human hand model into the scene.

The virtual hand system is built of geometric shapes, which recreate the hand bones anatomy. For each finger presented in Figure 2, we associate four cylindrical bodies, that are used for representing bones length and four sphere bodies, which connect them together, resulting the skeleton shape of the hand. Tracking algorithms interpret the data and deduce the positions of the undetectable hand elements from the Leap sensors, to ensure a continuous presence of the virtual hands on screen, as long as possible.

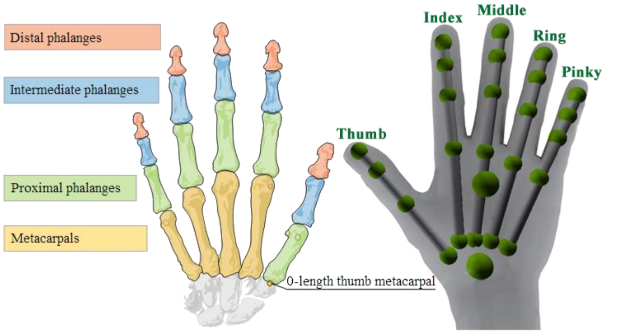


Figure 2. Hand anatomy [11] and virtual model used in app [12]

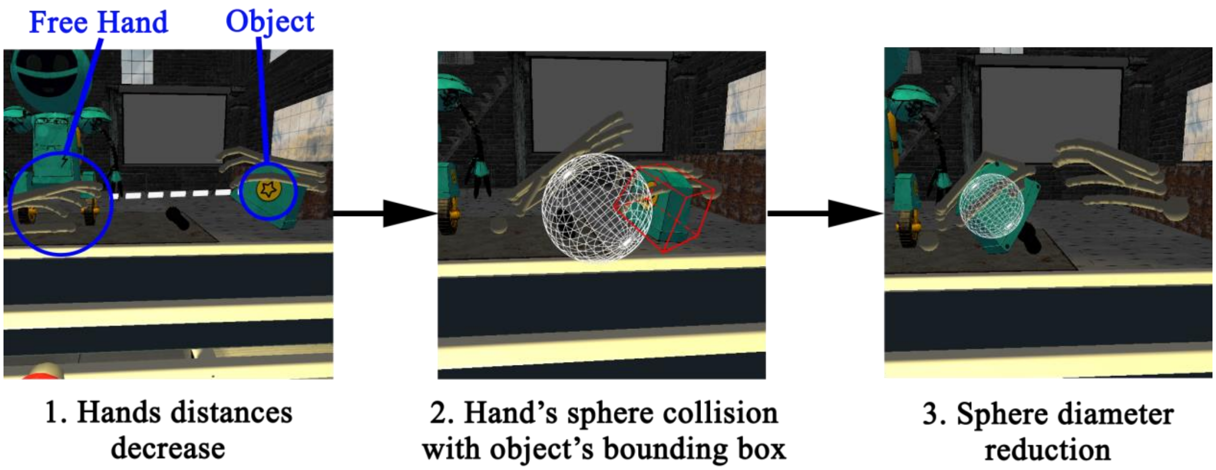

Free Hand
Object
1. Hands distances decrease
2. Hand's sphere collision with object's bounding box
3. Sphere diameter reduction

Figure 3. The steps of changing objects from one hand to another hand

The main code path of the VR application is executing a loop in which the Oculus camera sensors request the headset position, then creates the scene texture for each eye. The stereoscopic sensation is operated automatic by the Oculus SDK. The final rendered scene is post processed for each frame by Oculus Compositor, in order to apply distortion and then it is displayed onto the Rift's screen [13].

INTERACTION

Each device has their own coordinate system, which has to be synchronized, in order to be correctly displayed on the screen. Leap Motion tracking software processes the human hand on its visual angle, then Oculus library render the scene and place the virtual hand. model on its coordinate system. In order to use the LeapMotion device attached on the Rift headset, some operations are needed for placing the hand system in front of the virtual camera. As the Leap Motion company does not provide a mirrored hand system technique, all bones and joints have to be manually oriented, by flipping the hand information, received from Leap API, on the local Z axis [14]. Rotating the system at 90° on user's local X axis, will result in rendering the human hand motion in the intended place, similar to real interaction. These operations are also needed to be applied on objects, when interacting with them, in order to maintain the same coordinate system as the hand. When using LeapMotion device on a surface, there is no need of this correction anymore.

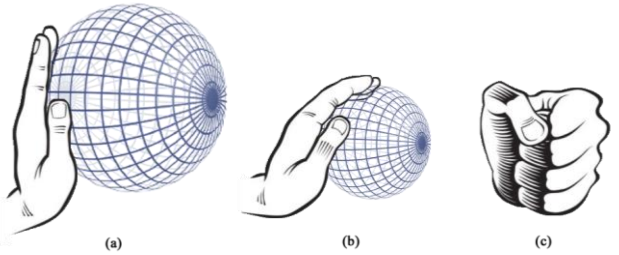


Figure 4. Hand elements of grabbing objects from scene, sphere diameter 180mm (a), ideal grab event 90mm (b), 0mm (c)

Leap SDK offers controller positions, rotations, normals and other data as 3D coordinates vectors. Hand rotation is computed based on hand direction, as the distance from the middle of the palm towards the fingers, and palm normal, as a vector pointing downward of the palm [15]. Based on palm's information, user can grab objects from the scene just by clenching their fist. For that, we create a virtual sphere which covers the length of the fingers, Figure 4. The sphere is placed roughly as if the hand was holding a ball. As fingers are closer to the palm, the sphere radius is reduced and when is used near an object, the grabbing event is triggered. The object is linked to the center of the sphere so that it gives the impression of holding it. In order to interact with a part of the assembly model, a free user hand has to be near the object. The sphere diameter is tested so as not to exceed a constant value, which matches the 45° hand angle. Touching the collision mesh of the object causes the link between hand and object. Both hands can be used simultaneously to interact with the scene and also to move objects from one to another hand, see Figure 3.

Complex gestures are available to be used by experienced users, to rotate model parts directly in the hand, without placing and grabbing them again from conveyor belt. This technique uses two hands, one is holding the object and the other one is performing gestures, in order to rotate the object based on the movement direction, Figure 5. To access free object rotation mode, user will perform a pinch gesture, with the other three fingers raised up, so that it is not interpreted like a grabbing gesture.

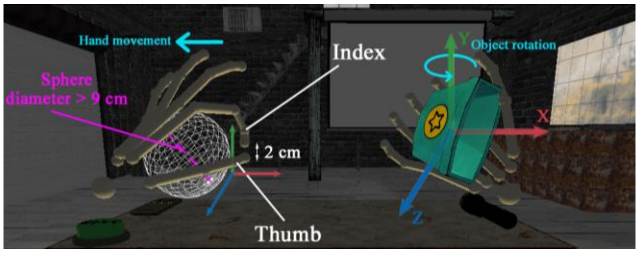


Figure 5. Complex gesture local coordinate system

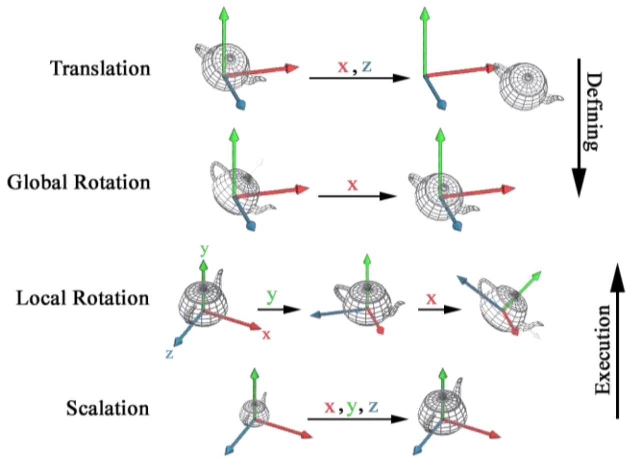


Figure 6. Object transformation order in application code

Overall, the object interaction is composed of 3 levels of rotation. When grabbing an object from scene, their coordinate system is attached to the corresponding virtual hand system, so that each tilt performed with the hand in any direction of the axis will be followed by the object. As the object can be grabbed from the conveyor belt in any direction user want, the rotation has to be made on global axis, after the local rotation computation, see Figure 6. All the operations made on the object are computed starting from the identity matrix, Equations 1 and 2, where m_{00}, m_{10}, m_{20} represent the X axis coordinate, m_{01}, m_{11}, m_{21} represent the Y axis coordinate, m_{02}, m_{12}, m_{22} represent the Z axis coordinate.

$$M_{identity} = \begin{pmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{pmatrix} \qquad (1)$$

$$M_{projection} = \begin{pmatrix} \boldsymbol{m_{00}} & \boldsymbol{m_{01}} & \boldsymbol{m_{02}} & m_{03} \\ \boldsymbol{m_{10}} & \boldsymbol{m_{11}} & \boldsymbol{m_{12}} & m_{13} \\ \boldsymbol{m_{20}} & \boldsymbol{m_{21}} & \boldsymbol{m_{22}} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \end{pmatrix} \qquad (2)$$

Beside their original importing process into scene, objects can be discarded in any posture when they touch the belt. The orientation is retained and recalculated throughout the entire running application, based on the final projection matrix of the object, Equations 3, 4 and 5, where α, β, γ are the roll, yaw and pitch angle rotations [16].

$$\alpha = arctg\left(\frac{m_{10}}{m_{00}}\right) * \frac{180}{\pi} \qquad (3)$$

$$\beta = arctg\left(\frac{-m_{20}}{\sqrt{{m_{21}}^2 + {m_{22}}^2}}\right) * \frac{180}{\pi} \qquad (4)$$

$$\gamma = arctg\left(\frac{m_{21}}{m_{22}}\right) * \frac{180}{\pi} \qquad (5)$$

The final layer of rotation is acquired by hand gestures, which is added to the local object rotation. The movement performed by the hand will be mapped on the next rotation axis of the object. An example is presented in Figure 5, when moving the left hand on X axis, will result in rotating the object around Y axis. This way, the object manipulation will rotate on the respective axis of the user performing the hand movement, making intended rotation behavior. All layers put together will result in a predictive system response to the human hands motion.

Beside their original importing structure into scene, objects are oriented based on the current state of rotation on the conveyor belt, LeapMotion sensor's horizontal angle of view and simple and complex hand interactions.

Algorithm 1. Hand-object matrix transformation

```
getObjectLocationAndOrientation()
.IdentityMatrix()
.translation(palmSphereCenter)
.rotation(90 * toRadians, AXIS(1,0,0))
.rotation(180 * toRadians, AXIS(0,0,1))
.rotation(pinchMovement.z, AXIS(1,0,0))
.rotation(pinchMovement.x, AXIS(0,1,0))
.rotation(pinchMovement.y, AXIS(0,0,1))
.rotation(hand.direction().pitch, AXIS(1,0,0))
.rotation(hand.direction().yaw, AXIS(0,1,0))
.rotation(hand.palmNormal().roll, AXIS(0,0,1))
.rotation(obj.rotation().x, AXIS(1,0,0))
.rotation(obj.rotation().y, AXIS(0,1,0))
.rotation(obj.rotation().z, AXIS(0,0,1))
.scaling(obj.scale())
.RenderModel()
```

CHILD OBJECTS

Interacting with a virtual world is not always friendly, because your real body is not transferred in the new environment. The presence of an avatar, which illustrate the human body, may give the impression of trust and also provides distance approximation of the virtual world. We attach a 3D model to the camera, so that looking down to the feet, the user will see parts of the model, see Figure 7 (a). Each user movement performed in real life will be followed in moving the virtual body model with the camera's position.

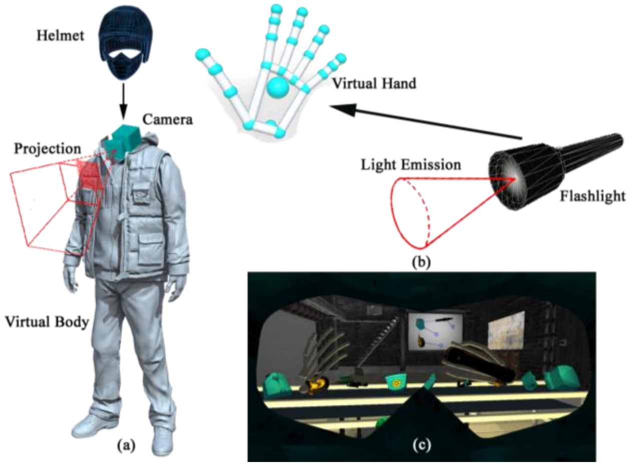


Figure 7. Child/Parent relations based on Oculus sensors (a) and Leap sensors (b). Final scene projection (c)

In the previous section we discussed about object interaction based on the virtual hand system, provided by Leap sensors, as shown in Figure 7 (b). Now, we will present a method of linking the objects to the virtual camera, based on Oculus sensors. Assuming we have a helmet, presented as a part of the model assembly, users may attempt to put it on them, finding themself inside the object. The virtual camera is now covered with the 3D model, which block a part of the eye visualization. Based on the headset's rotation on all the three axis, the object is now repeating the translation and orientation transformations after the camera's point of view, see Figure 7 (c).

VR headset orientation is provided by the Oculus library, in the right-handed cartesian coordinate system, stored in a quaternion. In order to use the same transformations technique as presented in Algorithm 1, we have to convert data orientation in Euler angles, which store all the X, Y and Z rotation angles in a vector, Equations 6, 7 and 8, where q_r, q_x, q_y, q_z represent the four quaternion elements and ϕ, θ, ψ are the yaw, pitch and roll angle rotations in radians [17].

$$\phi = arctg\left(2 * \frac{(q_w q_x + q_y q_z)}{1 - 2({q_x}^2 + {q_y}^2)}\right) \qquad (6)$$

$$\theta = arcsin\left(2 * (q_w q_y - q_z q_x)\right) \qquad (7)$$

$$\psi = arctg\left(2 * \frac{(q_w q_z + q_x q_y)}{1 - 2({q_y}^2 + {q_z}^2)}\right) \qquad (8)$$

The results are used to define helmet's correct orientation, then used together with camera's position in virtual world, we can render the 3D model in front of the camera.

Algorithm 2. Head-object tracking transformation

```
GetObjectLinkedToCamera()
.IdentityMatrix()
.translation(camera.postion())
.rotation(camera.rotation().pitch, AXIS(1,0,0))
.rotation(camera.rotation().yaw, AXIS(0,1,0))
.rotation(camera.rotation().roll, AXIS(0,0,1))
.translation(objectPositionOffset)
.scaling(obj.scale())
.RenderModel()
```

Naturally, light propagation or objects attached to other objects are following direct motion of the parent's model. For instance, placing a part of the assembly model in a pocket, will change its position based on the movement of the body. A light source emitted from a flashlight, which is hold by one of the hand has 2 levels of ascendancy.

MODEL BUILD

Model composition has steps and sometimes it has alternatives of the assembly process, by starting with base parts and finishing with covers, circuit boxes or supporting parts. The model building might be completed using parallel assembly, presented as a generic tree graph, which distributes the product parts in levels of requirements, Figure 8. The nodes represents the model parts and the connected networks define the attaching rules. Performing a level order traversal, we are making sure the correct workflow is provided.

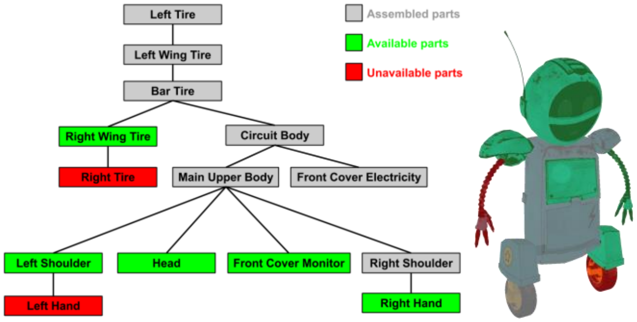


Figure 8. Assembly process based on N-array tree graph of the model parts

Creating an efficient mode of interaction with the objects, allow the user to stay still with the body. Model parts are split in hidden and visible objects, placed on a conveyor belt. Moving the belt forward or backward, will bring parts closer to the user and will make other parts visible to the camera's point of view. The assembly is constraints to all components, which makes the final model to be accomplished with all parts pieced together, which were initially on the conveyor belt.

Model creation

The engineering model has to be decomposed in parts and exported, using modelling tools, in the origin of the coordinate system. Based on object position and orientation in the final assembly form, the application can create events and correct areas of reconstruction in the virtual 3D world. All model parts are placed randomly on the conveyor belt and some rotation transformations are also applied to each of them. This way, the simulation will always be different. As not all the objects fit on the belt, some of them are excluded from the rendering process. Searching for other parts is done by sliding the belt in a certain direction, using directional buttons.

GUIDED ASSEMBLY

The application comes in many forms of learning, by evaluating the assembly performance, inspecting model parts, practice or testing the user composition capacity and problem solving skills. An instruction manual projected on a table might help people to identify the correct objects and accurately connect parts together, as shown in Figure 9. Novice users achieve experience by guided visual steps. All the possible parts are marked on screen, by creating a virtual bounding box model around the object, placed on conveyor belt. Grabbing the object will display the correct position and orientation on the assembly station.

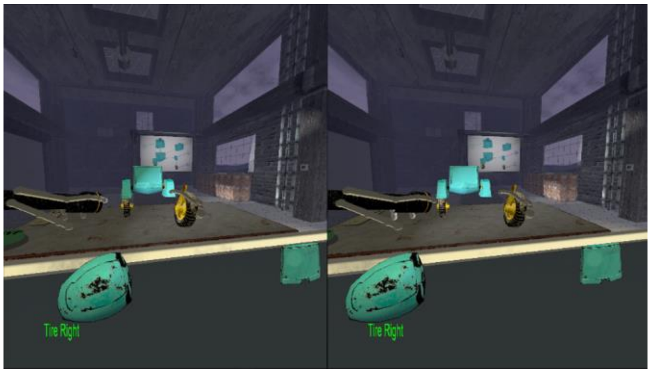


Figure 9. Virtual scene application split in two, corresponding to left and right eye

The Oriented Bounding Box (OOB) geometry mesh is computed using Vertex Buffer Objects (VBOs) by loading the lower and upper vertex limits of the object on all axis. The attributes are stored in memory as data vectors and therefore used to render the model behavior and properties on screen, like world position, texture coordinates, normals, colors, etc.

Some of the model parts might be similar, having the same structure, but different in terms of assembly steps. A labelling system is developed to show the object specifications, like name and metrics, so that users will be capable of associating easier and faster the parts, in order to achieve the final model. During the assembly session, users can check the parts properties from assembly station by passing the index finger through the model. Also this technique is used in triggering object events and buttons interaction.

PERFORMANCE

Performance is directly related with the human experience in using similar applications, motivation, attention to details and the level of knowledge. The first usage of virtual equipment by the user leads to accommodation with the system and adaptation with simple and complex interaction techniques available for use. In time, he might improve the experience, by training and by changing the application difficulty.

Other VR equipment uses physical buttons on hand controllers to activate interaction events. All buttons behavior are determined by the application and the interaction response might be defined as long-press or short-press of the button. Our solution set aside alternative touch interaction equipment, so that users are not learning new styles that are not related to real world interaction. Natural hand gestures detected by the LeapMotion sensor have the potential to be more natural and familiar than traditional methods. Based on hand gestures, the system analyzes the hand structure and performs one of the following actions: linking an object to the hand system, switching item's parent, releasing structure connection, running device response or enabling free object rotation. The amount of time spent by an user to understand the basic interaction features of the application is completed in a short period of time, achieved in the beginning of the simulation. To successfully make an action in the virtual world, the system leads the user intuition to try performing different gestures.

The assembly process requires users to have conceptual knowledge, analytical knowledge and metacognitive awareness. We did some tests on how much time it takes for the user to complete the reconstruction model of a robot composed of 14 parts, with both guided assembly methods combined and individuals. Five students and two employees aged between 21 and 35, who have not used a VR headset before, participated in this immersive assembly test. After some successful attempts, the following day, users were given the same model to assembly, without guidance elements. This way, we tested human retention and attentiveness. The final results are presented in Table 2.

Guidance	**Both methods**	**Interactive Indicators**	**Instr. Manual**	**None**
Completion Time (min.)	3:45	4:10	5:05	6:20

Table 2. Assembly performance time

SYSTEM REQUIREMENTS

The ideal virtual reality experience using DK2 is achieved with the maximum allowable latency of about 20ms and 75Hz application frame rate, based on the recommended hardware requirements for Oculus Rift. In this case, the whole scene is rendered in approximately 9ms and the rest of the time up to 13.33ms is allocated to image distortion for each frame. Once the double scene image is sent to headset screen, the system initializes the parameters for the next frame. The following software were used for the application implementation:

- Oculus Setup (latest version)
- Leap Motion Developer Kit version 4.0.0

The main libraries used are:

- Oculus SDK for Windows version 1.38.0
- Leap Motion Orion version 3.2.0

CONCLUSION

The assembly simulation using VR technology involves designing a model, breaking it down into pieces and developing a reconstruction process, based on linking rules. The purpose of this paper was to describe a virtual reality training tool of assembling a product in an immersive environment, based on human hand interaction. The engine developed processes the render algorithms, displays scene elements and computes 3D operations on objects based on user gestures. The overall solution inspired us to see many innovative ideas in which students and employees can benefit. Developing new learning tools by using current technology may affect future education system, so that the new generation to be more advanced on performing practical skills.

Future improvements include further refinement of hand gestures and may involve using special gloves equipment with wireless technology for interaction with the virtual world. The sensation would be much similar with reality by feeling that the objects are held in the user's hand, based on tiny vibrations applied to the fingers. Combining soft interactions, such as hand gesture and hard interactions, such as vibro-tactile feedback provides more natural interaction with virtual objects, similar to manipulation tasks. The result allows an improvement in the sense of presence perceived by the user during the interaction.

REFERENCES

1. Active Virtual Reality Users Forecast Worldwide, Statistica Research Department. https://www.statista.com/statistics/426469/active-virtual-reality-users-worldwide
2. Daniel Kersten, Pascal Mamassian and David C Knill, Moving cast shadows induce apparent motion in depth, *Perception, vol. 26* (1997), 171–92.
3. Zhao, Richard & Aqlan, Faisal & Elliott, Lisa & Lum, Heather, Developing a virtual reality game for manufacturing education (2019).
4. Laia Pujol-Tost and M. Phil, Interactivity in virtual and multimedia environments: a meeting point for education and ICT in archaeological museums, *University of the Aegean*, Department of Cultural Technology and Communication (2020).
5. Paul Zimmons and Abigail Panter, The Influence of Rendering Quality on Presence and Task Performance in a Virtual Environment, *Proceedings - Virtual Reality Annual International Symposium* (2003), 293-294.
6. Mihai Pop and Adrian Sabou, Gesture-based Visual Analytics in Virtual Reality, *Revista Română de Interacțiune Om-Calculator 10, Issue 3* (2017), 216–230.
7. Thomas Galais, Rémy Alonso and Alexandra Delmas, Natural Interaction in Virtual Reality: Impact on the Cognitive Load, *The Human Factors and Ergonomics Society / Europe* (2019).
8. A.C. Boud, D.J. Haniff, C. Babe, and S.J. Steiner, Virtual reality and augmented reality as a training tool for assembly tasks, *Proceedings of IEEE International Conference on Information Visualization* (1999), 32–36.
9. Patricia Baggett and Andrzej Ehrenfeucht, Building physical and mental models in assembly tasks, *International Journal of Industrial Ergonomics, Volume 7, Issue 3* (1991), 217–227.
10. Unity, Unreal, Godot, How to choose the best real-time 3D solution. https://www.ausy.com/en/technical-news/unity-unreal-godothow-choose-best-real-time-3d-solution-0
11. Kevin Horowitz, Skeletal Tracking 101: Getting Started with the Bone API. http://blog.leapmotion.com/skeletal-tracking-101-getting-started-with-the-bone-api-and-rigged-hands
12. Alex Colgan, Hand Hierarchy. http://blog.leapmotion.com/getting-started-leap-motion-sdk/hand-hierarchy
13. Oculus Documentation, Rendering to the Oculus Rift. https://developer.oculus.com/documentation/native/pc/dg-render
14. A. D. Bradley, B. Karen and A. B. Phillips, Oculus Rift in Action, *Manning Publications* (2015).
15. Computing the Hand Orientation. https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap_Hand.html
16. Steven M. LaValle, Planning Algorithms, *Cambridge University Press* (2006), 97-100.
17. Josè Luis Blanco Claraco, A tutorial on SE(3) transformation parameterizations and on-manifold optimization, *University of Málaga* (2020), 15–16.

Exploring the main factors driving a satisfactory use of the Moodle platform

Elena Ancuța Santi, Gabriel Gorghiu
Valahia University Targoviste
13 Aleea Sinaia, 130004 Targoviste, Romania
santi.anca@yahoo.ro, ggorghiu@gmail.com

Costin Pribeanu
Academy of Romanian Scientists
54 Splaiul Independentei, 050094 Bucharest, Romania
costin.pribeanu@gmail.com

DOI: 10.37789/rochi.2020.1.1.23

ABSTRACT

E-learning platforms are largely used in educational institutions around the world, in hybrid, mixed or blended learning formats. The power of those platforms has been proved extensively in the period of the *global pandemic* being used as main frameworks for offering and achieving educational services. This work is an exploratory study aiming to analyze the contribution of the perceived ease of use, usefulness, and enjoyment to the satisfaction of using a Moodle e-learning platform. To do this, a structural modeling approach has been taken. The model testing results show that perceived enjoyment was the most important factor which suggests that, apart from being usable and useful, a learning platform should be also attractive and interesting.

Keywords

e-learning, Moodle, TAM, usability, perceived enjoyment, user satisfaction.

ACM Classification

D.2.2: Design tools and techniques. H5.2 User interfaces.

INTRODUCTION

Modern academic education is increasingly capitalizing on the features provided by *Learning Management Systems* (*LMS*) or *Virtual Learning Environments* (*VLE*), as relevant software applications able to manage online teaching and learning. At the moment, being generically defined as "*platforms*", such applications are largely used in educational institutions around the world, in hybrid, mixed or blended learning formats, mainly combining face-to-face education (performed in the classroom) with the instruction offered in the online environment.

The power of those platforms has been proved extensively in the period of the *global pandemic*, when LMSs and VLEs constituted the main frameworks for offering and achieving educational services, exclusively online. In almost all the cases, related platforms began to be exploited near their full potential, starting with offering presentations and teaching materials for students to evaluating their knowledge by using various test formats or producing various statistics. In this respect, the ordinary face-to-face communication between teacher and student has been transposed in an environment very familiar to the actual generation of youngers, with impact on ensuring the proper pedagogical and psychological support, but also on maintaining the spirit of competition or modeling the student's personality [11]. In any case, among the many features of such platforms, Epping [9] specifies that each of them provides an environment for teaching and learning *without time or distance restrictions*.

Currently, there is a multitude of e-learning platforms, with different characteristics of structure, functionality, accessibility, and attractiveness. One of the most commonly used LMS is *Moodle - Modular Object-Oriented Dynamic Learning Environment*, with more than 156000 sites in 241 countries, as the situation is stated in July 2020 [20].

The aim of this research is to explore the contribution of the perceived ease of use, usefulness, and enjoyment to the satisfaction of using an e-learning platform, during the pandemic time, more precisely, the second semester of the academic year 2019-2020 (March - June 2020).

In section 2, related work is discussed with a focus on the analyzed factors. The method and sample are presented in section 3. Specific results are discussed in section 4. The paper is ending with a conclusion in section 5.

RELATED WORK

There are a series of studies in which the abovementioned factors were analyzed in different LMSs. In this respect, it can be started from Nielsen's remarks [15], who considered the product's usability as one of the main aspects that influence its acceptance by the specified users. To identify the best foldable model concerning the students' and teachers' educational needs, researchers in the e-learning area have studied various usability aspects of different platforms.

Aljawarneh [1] presented the main online learning platforms used in higher education, concluding that Moodle has proven to be the most effective in online learning, being preferred by many teachers and practitioners from all over the world. The actual statistics (July 2020), seem to be in-line with this [20]: 27 million courses are uploaded in various Moodle environments and 216 million users benefit of them and its advantages, also emphasized by Oproiu [16].

Bremer & Bryant [4] stated that 80% of students preferred *Moodle* vs *Blackboard*, being easier to use and having more tools that facilitate communication and collaboration. Pavlakou & Kalachanis [17] presented the main Moodle strengths and its efficiency in adult learning (life-long learning).

Even Dolendo [8], by comparing the usability of three LMS- *Moodle*, *Edmodo,* and *Schoology* - stipulated that Edmodo was the most usable LMS, the recent study conducted by Szirmai [18] highlighted the students' preference for learning when using Moodle platform, as a "*virtual extension of the teacher*".

Anarinejad & Mohammadi [2] and Usuf [19] revealed some strengths and weaknesses of the e-learning systems in higher education, but also Meiselwitz & Sadera [13] show that a connection exists among the usability and students' learning outcomes in online learning environments.

The study of Calli et al. [5] analyzed the influence of four factors on the user's satisfaction in e-learning systems. In their model, the usage intention is determined by the user's satisfaction which, in turn, is influenced by three variables: perceived usefulness, perceived playfulness, and content effectiveness. Perceived usefulness has three antecedents: perceived playfulness, perceived ease of use, and content effectiveness. The results show that perceived usefulness has the main influence on satisfaction and the perceived playfulness has the main influence on the perceived usefulness.

Pereira et al. [7] also analyzed the influence of satisfaction on the continued use of a web learning system. In their model satisfaction has two main drivers which manifest in several factors: performance (perceived quality, perceived value, quality, usability, and value) and technology readiness (optimism and innovativeness).

RESEARCH METHODOLOGY

Variables and hypotheses

This research is grounded in the Technology Acceptance Model (TAM) and its further extensions that posit that the perceived ease of use, perceived usefulness, and perceived enjoyment are the main drivers of the technology acceptance and continued use [6]. As many authors pointed out, satisfaction is an important variable for the intention to continue using a system [5, 7]. The following latent variables have been conceptualized and measured in this study: perceived ease of use (PEU), perceived enjoyment (PE), perceived usefulness of the platform (PU), and perceived satisfaction (SAT).

According to Davis et al. [6], the perceived ease of use refers to the belief that using a system is "free of effort". This means that the user will find it easy to understand, learn how to use, and operate a given system. The perceived usefulness is an extrinsic motivation that refers to the belief that using the system will "enhance job performance". The perceived enjoyment is an intrinsic motivation that refers to the belief that the system is "enjoyable in its own rights" [6].

The following hypotheses are tested in this study:

H1. Perceived ease of use has a positive effect on perceived enjoyment (PEU ⟶ PE).

H2. Perceived ease of use has a positive effect on the perceived usefulness (PEU ⟶ PU).

H3. Perceived ease of use has a positive effect on perceived satisfaction (PEU ⟶ SAT).

H4. Perceived enjoyment has a positive effect on the perceived usefulness (PE ⟶ PU).

H5. Perceived enjoyment has a positive effect on perceived satisfaction (PE ⟶ SAT).

H6. Perceived usefulness has a positive effect on perceived satisfaction (PU ⟶ SAT).

Platform and sample

In Valahia University, the courses on the Moodle platform are organized on different levels, for students registered at the license, master, doctorate, postgraduate or continuing education levels. At each mentioned level, the courses are structured by faculties. Within the faculties, the organization is done according to specializations.

Inside the platform, the student's area - after the authentication process - allows [11] to access the information uploaded for each course, to know the related tasks and deadlines, to view the course announcements, to know and manage the course calendar, to upload the own work in dedicated space or directory, to interact with the teacher by using synchronous and asynchronous channels (chat, forum), to know the results of the evaluation process.

To collect the measures, a questionnaire has been administrated to students in April 2020. After answering some general questions, the students have been asked to evaluate the items on a 5-points Likert scale.

A total of 155 students from Valahia University in Targoviste answered the questionnaire. After data screening for incomplete answers, 35 questionnaires have been eliminated so the working sample has 120 observations (76 females, 44 males). The age of participants is ranging from 20-29 years to 59-60 years. Most of them (85) have between 20 and 29 years old.

Preliminary testing of the model showed several low factor loadings so several items have been eliminated. The remaining observed variables are presented in Table 1.

Method

Based on the analysis from the literature [3, 10, 12], the following goodness-of-fit (GOF) measures have been used: chi-square, normed chi-square (χ^2/df), comparative fit index (CFI), standardized root mean square residual (SRMR), and root mean square error of approximation (RMSEA).

Data analysis was carried out using Lisrel 9.5 for Windows [14]. Model testing and validation were carried out following a two-step approach [3] including measurement and structural models.

ANALYSIS AND RESULTS

Descriptive results

The items, means, and standard deviations are presented in Table 1. All items have mean values greater than 3.50 showing a positive perception. Overall, the platform has been perceived as easy to use and useful, since the construct means are 4.03, respectively 4.04.

Data normality was investigated in terms of skewness and kurtosis. The values were all within the recommended level [12], supporting the moderate departure from normality for all variables.

Table 1. Variables

Item	Description	M	SD
PEU1	How to use the platform is clear and intuitive	3.91	0.80
PEU2	The instructions on how to use this platform are well organized	4.04	0.84
PEU3	Finding information on this platform is easy	3.97	0.91
PEU4	The terms used in the user interface are clear	4.18	0.80
PE1	I like to use this platform	3.81	0.94
PE2	Working with this platform is attractive	3.71	1.04
PE3	The educational activity on this platform is interesting	3.80	0.99
PU1	The platform provides useful information on the progress of my work	4.01	0.88
PU2	The platform is useful for the communication student-teacher	3.85	1.01
PU3	The platform is useful for online educational activities	4.26	0.87
SAT1	This platform satisfies my needs for online education	3.73	0.98
SAT2	I am satisfied with the results I got working on this platform.	3.88	0.92
SAT3	I am satisfied with educational activity on this platform	3.69	0.95

Measurement model

Convergent and discriminant validity of the model has been analyzed using the procedure of Fornell and Larcker [10].

All standardized factor loadings are statistically significant. The item reliability (R^2) values are ranging between 0.60 and 0.92, above the suggested standard of 0.50 [12]. Cronbach's alpha values are acceptable for all three constructs (see Table 2).

The composite reliability (CR) values ranged from 0.807 to 0.909, above the minimum level of 0.70 [12]. The values of the average variance extracted (AVE) ranged from 0.514 to 0.738, all above the minimum level of 0.50 [12], thus confirming the convergent validity of the model.

The discriminant validity of constructs was examined through the squared correlations test [10]. The results in Table 2 show that with two exceptions, the square root of the AVE for each construct is greater than the correlations between constructs, which shows acceptable discriminant validity.

Table 2. Results of convergent and discriminant validity

	Alpha	CR	AVE	PEU	PE	PU	SAR
PEU	0.81	0.807	0.514	0.717			
PE	0.93	0.930	0.816	0.574	0.903		
PU	0.78	0.788	0.555	0.738	0.699	0.745	
SAT	0.91	0.909	0.769	0.577	0.895	0.727	0.877

Notes: The bold diagonal numbers are the square root of AVE

Structural model

Structural equation modeling (SEM) was carried on to test the fit between the research model and the data. The structural model in Figure 1 shows the standardized path coefficients and the item loadings.

The model estimation results showed that PEU has a significant positive influence on PE (β=0.574, p<0.001) and PU (β=0.502, p<0.001) so the hypotheses H1 and H2 are supported. The influence of PEU on SAT is not significant so H3 is rejected. The path from PE to PU is also significant (β=0.410, p<0.001) so H4 is supported. Both PE and PU have a significant effect on SAT (β=0.70, p<0.001, respectively β=0.24, p=0.015), providing support for hypotheses H5 and H6.

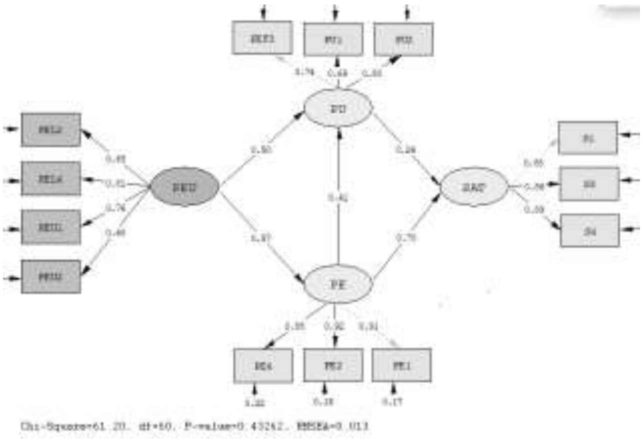


Figure 2. Structural model

The path coefficients show that the perceived enjoyment has a more important contribution than the perceived usefulness to the satisfaction with the platform. This is surprising since the e-learning platform is a pragmatic rather than a hedonic technology.

The model explains 33% of the variance in the perceived enjoyment, 65.7 % in the perceived usefulness, and 77.9% in the satisfaction with the platform.

The structural model fits very well with the data. The χ^2 test is nonsignificant (χ^2=61.20, *df*=60, *p*=0.433). The other GOF indices are also indicating a very good fit: χ^2/df=1.02, CFI=0.999, GFI=0.93, SRMR=0.030, RMSEA=0.013.

DISCUSSION

This study contributes to a better understanding of relationships among those factors that influence students' perceptions concerning the usability of the Moodle platform as an efficient e-learning tool. The resulting model provides a good understanding of those variables, in order to improve teacher intervention.

From the psychological perspective, the Moodle platform's *ease of use* (*PEU*) represents a feature that does not force the student cognitive resources. In this sense, the used and learned algorithms gradually become automatisms and do not require an increased student's effort, contributing also to achieve a certain level of *efficiency* (*PU*) in the activity carried out (quantified in results), but leading to a perceived *enjoyment* or level of *attractiveness* (*PE*) related to the product. This perceived enjoyment mobilizes or energizes the learning activities, contributing to the increase in work efficiency. Practically, this is how the inner motivation for using such a platform appears.

Although the efficiency contributes to obtaining the satisfaction to a certain extent, the results show that perceived enjoyment (PE) influences satisfaction (SAT) to a much greater extent. The measures of this construct are pointing to three attributes of the learning process: pleasant, attractive, and interesting. Perceived enjoyment has both a direct influence and an indirect influence mediated by the perceived usefulness.

The results are somehow surprising given the pragmatic nature of the platform and show that the attractiveness of a product can generate satisfaction to a greater extent than its efficiency, so how a product/content is presented must be taken into account.

The perceived ease of use has only an indirect influence on the satisfaction that is mediate buy the perceived enjoyment and perceived usefulness.

There are inherent limitations of this exploratory study. First, the sample is small (120 observations). Secondly, only three factors have been considered in this research. Future work should extend the scale by taking into account external variables, such as e-learning content.

CONCLUSION

The results obtained from the investigation concerning the usability of the Moodle platform illustrate that the respondents considered e-learning an interesting, pleasant, and efficient activity, which generates satisfaction and confidence in their ability to support learning. The strengths of the platform - *ease of use*, *productivity*, and *facilitator of learning* - have been highlighted by the students, most of them being convinced that they easily transfer their acquired skills in other distant learning situations if necessary. The results suggest that learning should be both attractive and interesting, but also find Moodle as a virtual environment proper to be exploited in the academic field, being very accessible to students.

REFERENCES

1. Aljawarneh, S. A. (2020). Reviewing and exploring innovative ubiquitous learning tools in higher education. *Journal of Computing in Higher Education*, *32*, 57-73, DOI: 10.1007/s12528-019-09207-0.
2. Anarinejad, A., Mohammadi, M. (2020). The Practical Indicators for Evaluation of E-Learning in Higher Education in Iran. *Interdisciplinary Journal of Virtual Learning in Medical Sciences*, *5*(1), 11-25.
3. Anderson, J.C., Gerbing, D.W. (1988). Structural Equation Modelling in Practice: A Review and Recommended Two-Step Approach. *Psychological Bulletin 103 (3)*, 411-423.
4. Bremer, D., Bryant, R. (2005). A comparison of two learning management systems: Moodle vs Blackboard. *Proceedings of 18th Annual Conference of the National Advisory Committee on Computing Qualifications*, Tauranga, New Zealand, 135-139.
5. Calli, L., Balcikanli, C., Calli, F., Cebeci, H. I., Seymen, O. F. (2013). Identifying factors that contribute to the satisfaction of students in e-learning. *Turkish Online Journal of Distance Education*, 14(1), 85-101.
6. Davis, F. D., Bagozzi, R. P., Warshaw, P. R. (1992). Extrinsic and intrinsic motivation to use computers in the workplace. *Journal of Applied Social Psychology*, 22(14), 1111-1132.
7. de Melo Pereira, F. A., Ramos, A. S. M., Gouvêa, M. A., & da Costa, M. F. (2015). Satisfaction and continuous use intention of e-learning service in Brazilian public organizations. *Computers in Human Behavio*r, 46, 139-148. DOI: 10.1016/j.chb.2015.01.016.
8. Dolendo, M. E. (2016). Usability Measurement of Learning Management Systems: A Response to Educational Technology Influx. *GSE E-Journal of Education*, *4*, 25-36. http://worldconferences.net/home 25.
9. Epping, R. J. (2010). Innovative Use of Blackboard [R] to Assess Laboratory Skills. *Journal of Learning Design*, *3*(3), 32-36. https://www.learntechlib.org/p/ 55583.
10. Fornell, C., Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research*, *18*(1), 39-50.
11. Gorghiu, G., Bîzoi, M., Gorghiu, L. M., Suduc, A. - M. (2009). Aspects Related to the Usefulness of a Distance Training Course Having Moodle as Course Management System Support. *Proc. WSEAS International Conference on Distance Learning & Web Engineering*, Budapest, Hungary, 54-59.
12. Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. (2006). *Multivariate Data Analysis*. 6th Ed., Prentice-Hall
13. Meiselwitz, G., Sadera W. A. (2008). Investigating the Connection between Usability and Learning Outcomes in Online Learning Environments. *MERLOT Journal of Online Learning and Teaching*, *4*(2), 234-242.
14. Mels, G. (2006). *LISREL for Windows: Getting Started Guide*. Lincolnwood: Scientific Software International Inc.
15. Nielsen, J. (1994). Heuristic Evaluation. In J. Nielsen & R. L. Mack (Eds.). *Usability Inspection Methods*. New York, NY: John Wiley & Sons
16. Oproiu, G. C. (2015). A Study about Using E-learning Platform (Moodle) in University Teaching Process. *Procedia - Social and Behavioral Sciences*, *180*, 426-432.
17. Pavlakou, E., Kalachanis, K. (2018). Adult Education Using the Moodle E-Learning Platform: The Role of the Trainer. *Journal of Education & Social Policy*, *5*(3), 105-109. DOI: 10.30845/jesp.v5n3p13.
18. Szirmai, M. (2020). Moodle: The Ubiquitous Teacher. *Electronic Journal of Foreign Language Teaching*, *17* (1), 190-204.
19. Usuf, B. N. (2020). Are we prepared enough? A case study of challenges in online learning in a private higher learning institution during the COVID-19 outbreaks. *Advances in Social Sciences Research Journal*, *7*(5), 205-212. DOI: 10.14738/assrj.75.8211.
20. ***, Moodle statistics. https://moodle.net/stats

Student Testing Activity Dataset from Data Structures Course

Paul Stefan Popescu, Marian Cristian Mihaescu, Oana Maria Teodorescu, Mihai Mocanu

University of Craiova

Department of Computers and Information Technology

{spopescu, mihaescu}@software.ucv.ro, teodorescuoanamaria@yahoo.com, mocanu@software.ucv.ro

DOI: 10.37789/rochi.2020.1.1.24

ABSTRACT

E-learning platforms became more and more popular not only for distance learning but also for learning in full-time education. As this popularity grows, we can use the data extracted from them to complement the professor's work and make predictions regarding students' performance. In this paper, we present a dataset extracted from our e-Learning platform, which is based on the logs collected from testing activity. The focus of this paper is to present the dataset; the experiments presented in the paper are meant to explore the dataset along with its capabilities. The dataset consists of attributes relevant to the testing activity and provides labels which consist of average test grade and final exam grade. Our focus when building the dataset was to keep only the attributes relevant for the learning activity and to provide means to analyse and predict the student's final grade or failure. The paper presents the structure of the dataset, the methodology of collecting the data and experiments using several popular algorithms. The experimental results reveal that the actions performed by the users correlate with the results of the tests and the exam failure can be predicted with a pretty good accuracy using the default set of tuning parameters for our algorithms. As feature work, we can extend the set of experiments with other algorithms, and we can also use parameter tuning for each algorithm for a slight increase in performance.

Author Keywords

Machine Learning, Educational Dataset, e-Learning, User Modelling

ACM Classification Keywords

H.5.m. Information interfaces and presentation: Miscellaneous.

INTRODUCTION

This paper refers to the area of educational datasets which can be used for machine learning tasks. Even if the research area of educational data mining started a long time ago, there are few public educational datasets which can be used as a reference when classifying or predicting the student's performance. The main problem when referring to the prediction of student's performance is a lack of a generic and reproducible approach which may lead to a relevant and generic result. We aim to produce a dataset which can be continuously updated and can help instructors to early predict student's failure using machine learning algorithms.

In this paper, we present a dataset extracted from the testing activity logged in our e-Learning platform which offers all the required functionalities for online education. The features extracted for this dataset are referring the testing activity and they can be gathered from most e-learning platforms available these days. After presenting each feature of the dataset along with several metrics which can provide a deeper understanding of the data, we conduct some experiments using some common machine learning algorithms. Even though we use an e-Learning platform for taking tests, students who contributed with their actions are from full-time education programs and taking tests was part of the requirements of the Data Structures course in which they were enrolled.

The e-learning platform was custom implemented for running at our University mainly for research purposes and has four roles implemented: student, professor, secretary and administrators. The main functionalities of the platform are learning resources management, communication (between students, students and professors and students and secretaries), testing and live presentations. The learning resources management module offers courses management, homework and external references; the course management option includes the testing setup, which makes the subject of this paper. The platform was designed to be easy to use and to log most of the user actions in order to provide useful insight regarding the activity performed and also to provide relevant research data.

Despite the differences between our platform and other online educational environments, the main features of the platform are quite generic and the structure of the dataset can be obtained from most of the other platforms, so the results are relevant for a large number of researchers. The novelty that comes with this dataset is that we logged all the information regarding the testing procedure, so the grade which is also the class (or the value we want to predict) benefits from several relevant attributes. Another aspect regarding the class attribute is that we computed the mean grade obtained from the tests, which can be one of the classes. However, we also added the grade obtained at the final exam so we can analyse and compute how much taking tests influences the final grade.

The primary motivation of computing and publishing such a dataset is that we can predict the student's failure before the exam will take place even if this is not an evolutionary approach. The reason is that in our scenario, the students took tests before the semester ended and they had to take the exam after that so that the test's grade may be a relevant indicator for their final result. There is also a correlation between the tests they took and the final exam as the questions used in the tests referred to the same topics they have to learn for the final exam. Even though the topics were the same, there are several differences between the tests they took and the final exam as the tests have questions from only a part of the topics necessary at the exam.

RELATED WORK

One of the most used and referred datasets in educational data mining literature is [1] which logs 30 attributes about students and offers three grades: G1, G2 and which corresponds to the grades obtained during first and second periods (or semesters) and G3 which is the final grade. The primary dataset which is stored at UCI Machine Learning Repository consists of two csv files which can be used together or separately; one of it logged the data for a mathematics course, and the other one logged the data from a Portuguese course. The main catch of the dataset is that it offers many attributes, mostly demographic about the students and final grades, without giving a better insight into how well a student performed in the educational environment. Computing demographic attributes may offer a better insight about student's personality or situation, but in terms of education, it may not reflect its focus or how well the student it is engaged in the activity.

Although the dataset is useful for final grade prediction and it is used in many papers, some entirely new [2] but, it is impossible to predict the final grade at early stages and to prevent the failure. Another problem of this accessible dataset is that it does not offer a good classification accuracy for the final grade or the other good grades. For both of the datasets (mathematics and Portuguese), you will hardly get any better than 50% accuracy. Depending on the data preprocessing the analyst can obtain better results like in [3] or [4] but still the last grade (G3) depends on G1 and G2.

Another popular dataset is presented by Amrieh et. al. in [5] and previously in [6] and it's available on Kaggle platform. The data is gathered from a learning management system called Kalboard 360 using a learner activity tracker tool which is called experience API (xAPI). The xAPI tool is a component of the training and learning architecture that allows tutors to monitor the learning progress and actions like reading an article or watching a video. The authors collected a variety of features divided into four categories: demographic features (4), academic background features (6), parents’ participation on the learning process (2) and behavioural features (4). Initial experiments conducted on the 480 instances dataset reveal a good accuracy which varies from 70% to 80% as the authors state in the paper. The dataset was used in many papers with slightly better results for classification [7] or clustering [8] tasks.

There are also newer but not so popular datasets like *Open learning analytics dataset* which is described in [9] and discussed [10]. The dataset consists of several *.csv* files which describe tables from the database like courses, assessments, studentInfo, available materials and their relations. In this case, we do not have a dataset with a specific number of instances because it depends on which tables we want to merge, but the number of students that contributed to the dataset is huge. In studentInfo.csv file, there are 32593 recordings, each of them having a column for the final result which can be used as a class.

Another newer dataset is offered by Duolingo [11] which aims for a shared task on second language acquisition modelling [12] and they also launched a competition regarding this dataset. Regarding the competition, participants receive an English sentence and have to produce a high coverage set of translations in the target language. In order to level the playing field, the authors also provide a high-quality automatic reference translation (via Amazon), which may be considered as the baseline for the machine translation task. The data offered for the task comes from five Duolingo courses. All use English prompts, with multiple translations, although weighted by frequency from speakers of each of the following languages: Portuguese, Hungarian, Japanese, Korean and Vietnamese.

The world's largest repository of learning interaction data is PLSC Data shop [13], which offers a significant amount of learning data. In [14] the repository is well explained and in preset, there is plenty of data which can be queried and used for analysis. Still on the area of big educational data is ASSISTments Ecosystem [15] which is a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching but both of this systems it is a big difference in producing a dataset comparing to downloading one from Uci machine learning repository.

STRUCTURE OF THE DATASET

The dataset consists of eleven attributes which describe the student activity performed by 275 students (instances) performed during tests. We consider a number of eleven attributes even if in the list in which we present them we count thirteen attributes because MeanTestGrade is the same but in two versions: with continuous values and with discretized values and the exam grade also have two versions: with exact values obtained at the exam and with two values (0 and 1) which signals failure or success.

The attributes are relevant for the activity performed during the testing period, and the data was collected in two distinct years of study: 2018 and 2019. The dataset is publicly available on Kaggle[1] and can be used for further experiments.

1. *NumberOfLogins* - the number of logins made by the student during the testing period
2. *TimeSpentOnPlatform* - the total time spent on the platform by the student
3. *NumberOfTests* - the number of tests the students took
4. *TimeSpentForTests* - the total time spent by the students to take tests
5. *AverageTimePerTest* - the average time spent for taking one test
6. *NumberOfConcepts* - the total number of concepts included in his tests
7. *NumberOfActions* - the total number of actions logged for the student
8. *NumberOfRevisions* - the number of times student revised the questions from past tests
9. *LastGrade* - the grade obtained at his last test
10. *MeanTestsGrade* - the average grade obtained from all tests
11. *MeanTestsGradeD* - the same average grade but discretized with values from 4 to 10
12. *ExamGradeD* - the final grade obtained by the student at the final exam which is a discretized value
13. *PassExam* - the final grade divided into values: 0 for failure and 1 for passing the exam

Table 1. Attributes statistics summary

Attribute	Min	Max	Mean	StdDev
NumberOfLogins	1	42	7.57	5.74
TimeSpentOnPlatform	2	948	188.95	160.18
NumberOfTests	1	28	8.5	5.39
TimeSpentForTests	0	122	37.44	23.47
AverageTimePerTest	0	9.1	4.14	1.7
NumberOfConcepts	0	11	9.96	2.62
NumberOfActions	5	330	74.67	49.6
NumberOfRevisions	0	72	12.25	10.46
LastGrade	1	10	6.67	2.39
MeanTestsGrade	1.45	9.77	6.41	1.7
MeanTestsGradeD	4	10	NA	NA
ExamGradeD	3	10	6.44	1.82

Table 1 presents the attributes and a short analysis regarding their values. We focus on minimum, maximum and mean values along with standard deviation, which is presented in the last column because most of the values are numeric; these metrics are relevant in this case. Attribute 11 from the table have discrete values, and we cannot compute the mean and standard deviation, and this is why we have NA on those columns. The number from the first column corresponds to the attribute id from the previous list of items.

Each of the above-presented features is relevant for predicting the final grade and student's engagement in the learning activity. The number of logins along with time spent on platform and number of actions are influencing the grade as a quantity metric on how much the students are engaged in a learning activity through the platform as a bigger value implies more engagement. Usually a more engaged student will also be interested in gaining better grades and knowledge improving.

The number of tests is relevant for the dataset and the final grade because it is a good indicator on how good the student's implication is in the learning process and also how big is the influence of the other actions on the final result.

The number of concepts addressed by the student are relevant because they are a good indicator of how good the student's progress is. The relevance comes from the recommender system implemented in the platform, which allows the student to get question-related to a concept only after passing a certain threshold which signifies that he knows very well previous concepts.

The number of revisions refers mainly to how many times a student accessed a past test in order to see which questions were correctly answered and which was the correct answer. This concept is relevant for both learning engagement and predicting the final grade because a higher interest in the past tests means that the student aims for better results, and he wants to improve his knowledge. This feature, itself, can provide a significant insight regarding the student's behaviour because there are several cases: have a high grade and revise the questions, have a high grade but omit the revising and then the other two: have a small grade and revising and have a small grade but not revising the answers. These four situations need to be also addressed as future work as they provide valuable data which analysed can help the student's modelling and improve the grade or failure predicting.

[1] https://www.kaggle.com/cristianmihaescu/dsa-test-dataset

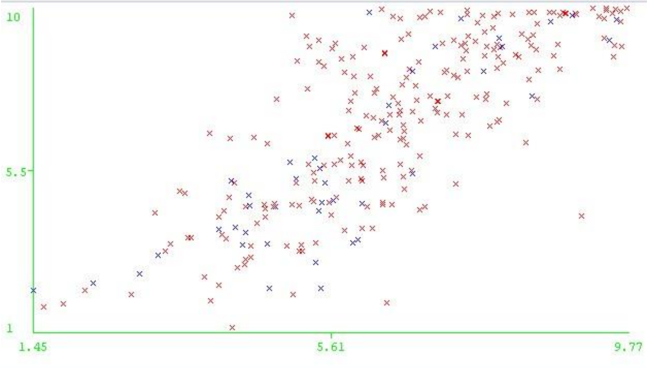


Figure 1. Last Grade vs MeanTest Grade Correlation

The last grade is relevant for the final grade or the student's level of knowledge because it marks at what level the student stopped taking tests and another motivation for computing this attribute is the correlation between it and the final result. In Figure 1, there is the correlation between *LastGrade* on the OY axis and *MeanTestGrade* on OX axis. It is visually clear that there is a correlation between these two attributes. The colour of the points for the pic corresponds to the final grade, and we have blue for fail and red for passing.

The mean grade for tests is presented in two ways, the average grade computed from all the tests and a discretized version. The grade discretization is made considering that what is more significant than 0.5 points we consider to be the next grade and what is less means the integer. For example, 4.3 will be considered as four while 4.7 will be discretized as a 5. The first version of the grade is more accurate because the value of the mark was not approximated, but it limits us to mainly regression algorithms, but discretized version allows us to use a greater variety of algorithms like classification algorithms.

Exam grade was already discretized, and it is the grade obtained by the student at the final exam. Trying to predict the exam grade based on the previous features, including the mean test grade is an excellent way to predict the student's failure. The mean test grade can also be a good estimator for the exam failure, and we aim to prevent it from making recommendations to students based on their testing results.

The motivation for adding the *PassExam* attribute can be deducted from analysing the grades distribution. Based on the grades obtained at the exam we have seven classes from which the only one is for failing the exam, and the rest of six is for passing. Dividing the data into two classes reduces the information gain split and reduces the problem to binary classification even if we must deal with imbalanced classes.

METHODOLOGY USED FOR COLLECTING THE DATA

The data was collected during the Data Structures course and is based on the graphs related topics. Graph topics represent half of the Data Structures course and are taught in the middle of the semester so deepening this part will have a significant impact on the final grade. Another benefit is that if we can predict the student's failure during this period, there is still enough time to catch up so building a dataset along with a system which can trigger a failure alert can be very beneficial for students who are taking this course.

The flow for constructing the dataset starts from the student who takes tests, then these tests are logged in the e-Learning platform, and after the testing period is finished, we can export the logs saved in the database and feed them to the dataset generator tool. The tool is able to execute queries on the database, computes the features for each student and build the dataset which can be used for signalling failure tot the student, to provide feedback regarding the knowledge level or to predict the exam result.

The testing period started after the graph’s chapters were taught and the student has enough knowledge to answer the questions. The testing procedure is based on a concept map which is represented by a directed acyclic graph. Students had to take at least five tests from the graphs in order to have a testing grade computed and to be included in the dataset. The number of questions for each test varies from eight to ten because we logged in this dataset two years of studies and in the first year we considered ten questions per test, and in the next year, we decreased the number of questions. The reason for decreasing from ten to eight questions per tests is that in many cases, students were not able to complete the questions from the last concepts of the graph

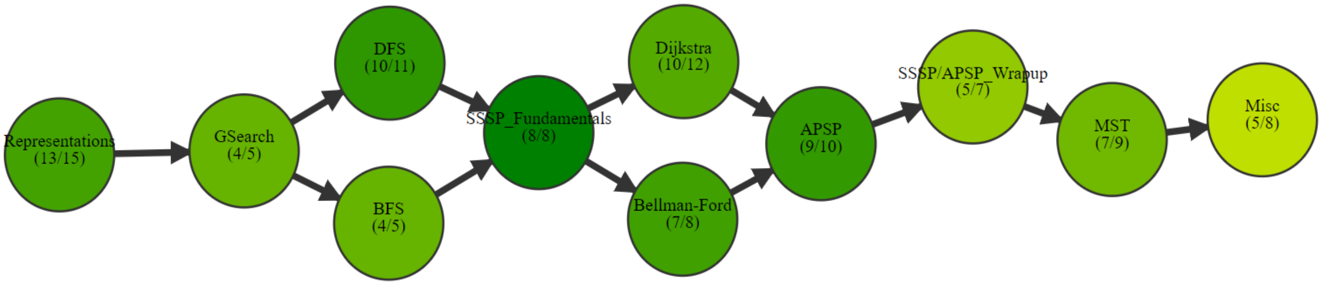


Figure 2. Graph example

Table 2. Concepts and questions per concept

Concept	No. of questions
Representations	15
GSearch	5
DFS	11
BFS	5
SSSP-Fundamentals	8
Dijkstra	12
Bellman-Ford	8
APSP	10
SSSP/APSP-Wrapup	7
MST	9
Misc	8

Table 2 presents the concepts along with the number of questions allocated for each concept. There are 98 concepts distributed for 11 concepts, and their dependency graph is presented in Figure 2. The succession of the concepts in the graph corresponds to the succession during the semester, so first they will learn *Representations*, then *GSearch* and so on. The first test which will be held will have questions from *Representations* concept and then and then after the student respond very good to the questions from this concept, he will advance to *GSearch*. The threshold for advancing from a concept to another is 50% and to have a concept accomplished is 75%, so a student that took tests from *GSearch* responded correctly to at least 50% of the questions form *Representations* and in order to skip getting questions from *Representations* he must have answered correctly to more than 75%.

EXPERIMENTS

This section presents a set of experiments conducted using this dataset. Our aim, in this case, is to explore the dataset and to investigate if we can predict the tests grade and the final result based on the logged data. A good correlation between features computed for the dataset and any of the labels will make the features we computed good predictors for the final result or the results of the test. The experiments are done using machine learning, having regression and classification tasks. For conducting experiments on this dataset, we used Weka [16] machine learning library which offers a easy to use variety of algorithms. For all the algorithms used in our experiments we considered the default values for the tunning parameters offered by Weka. Two other kernels which can be used for further and more complex experiments are available on Kaggle[2].

Table 3. Feature importance

OneRAttributeEval		InfoGainAttributeEval	
Exam	Test	Exam	Test
2	9	9	9
8	5	4	8
9	1	2	4
3	6	3	7
1	8	5	1
6	3	8	5
7	4	6	2
4	2	7	3
5	7	1	6

In order to get a better insight regarding the features and what is their importance in the dataset we use *OneRAttributeEval* and *InfoGainAttributeEval* algorithms with *Ranker* search method. There are several methods which can be used to compute the feature importance in the dataset because the feature importance depends on how we construct the model. We consider two different algorithms to analyse the dataset and see which features play a big role in the classification process. For both the algorithms we used the *MeanTestGradeD* and *Failure* attributes to evaluate the dataset so Table 3 presents on the first column the feature importance for *InfoGainAttributeEval* algorithm used on *Failure* attribute because we use *Failure* to evaluate the exam grade and on the second column the evaluation performed on *MeanTestGradeD* attribute. The same follows for the *InfoGainAttributeEval* algorithm and the values represent the attribute indices from the items list.

The attributes are included in Table 3 into descending order of their importance. Hence, the most important attribute in the dataset is the one from the first line, and on the last line, we have the least important attribute. One first conclusion when analysing the table is that the last grade obtained at the tests have a significant impact on the information gained and the number of revisions of the past tests are also important.

[2] https://www.kaggle.com/cristianmihaescu/dsa-test-dataset/kernels

Predicting the final tests result

First part of the experiments focus on predicting the test results and analysing the correlation between the logged data and the tests results.

Table 4. Results obtained for regression task

Algorithm Name	Correlation coeff	RMSE
SimpleLinearRegression	0.7719	1.0793
LinearRegression	0.782	1.0582
SMOReg	0.7805	1.0647
AdditiveRegression	0.7359	1.1631
RegressionByDiscretization	0.6875	1.2917
GaussianProcesses	0.5961	1.3816
RandomForest	0.7775	1.0704
RandomTree	0.6032	1.5254
DecisionStump	0.67	1.2604

Table 4 presents the results for a selection of algorithms used on the dataset for the regression task. On the first column of the table, we have the algorithm name as it appears in Weka, on the second column we have the correlation coefficient and on the final column we have RMSE which stands for Root Mean Squared Error metric. The best result was obtained by Linear regression with a correlation coefficient of 0.782 and an RMSE of 1.0582. For this case, we eliminated the final grade from the dataset and also the discretized version of the grade.

Table 5. Results for classification task using DT

Algorithm Name	Accuracy	RMSE
J48	36.36	0.36
DecisionStump	24.72	0.30
LMT	41.45	0.29
HoeffdingTree	37.45	0.33
RandomForest	40.36	0.3
RandomTree	31.63	0.4
REPTree	34.18	0.31

Table 5 presents the results obtained for classification tasks using decision trees algorithms. In this case, we used the discretized version of the final testing grade. The results, in this case, are not great because most of the features are numeric and the label is a nominal value. We included all the decision trees along with some ensembles because in most of the cases decision trees offer comprehensible models which are valuable for analysis of educational data. In the case of the Random Forest algorithm, which was the algorithm that outperformed the others, we used 100 iterations for training.

Table 6. Classification using various algorithms

Algorithm Name	Accuracy	RMSE
LogisticRegression	43.63	0.29
NaiveBayes	24.72	0.30
BayesNet	37.81	0.31
AdaBoostM1	24.72	0.30
DecisionTable	36.72	0.30
ZeroR	24.72	0.32
OneR	38.9	0.39

Table 6 continues the experiments conducted with discretized values for the final testing grade, and we still cannot get good results so predicting the grade of the test based on the actions performed during the testing period is a difficult task. In both tables 5 and 6, we kept only the discretized version of the grade and eliminated the final result along with the continuous version of the testing grade.

Predicting the student's failure

Considering a scenario in which student finished the graph testing period and we want to prevent failure at the final exam we divided the final grade into two classes: 1 and 0: 1 for passing the exam and 0 for a possible failure. In this section, we aim to explore how most common machine learning algorithms work on this dataset for predicting the student's failure. This task is relevant for the dataset as the testing period ends before the exam period and there is still time to trigger the alarm regarding student's final result at the exam and make recommendations regarding their activity.

Table 7. Results for predicting student’s failure

Algorithm Name	Accuracy	RMSE
J48	80.72	0.39
RandomForest	82.90	0.37
RandomTree	79.27	0.45
LMT	84.36	0.36
DecisionStump	83.27	0.37
HoeffdingTree	84.36	0.36
REPTree	83.63	0.37
ZeroR	84.36	0.36
OneR	84.36	0.39

JRip	82.18	0.37
Logistic	84.36	0.36
NaiveBayes	72.36	0.46

Table 7 summarises the results for predicting the student's failure. In the first part of the table, several decision trees algorithms are presented, and after that, a selection of several other algorithms. An accuracy of 84.36, which is also the best accuracy obtained in this case, is obtained by several algorithms and overall accuracy is above 80% correctly classified instances. This accuracy is consistent over several algorithms from different classes, and it is obtained just running the algorithms without doing any parameters tuning or applying features engineering techniques

CONCLUSIONS AND FUTURE WORK

In this paper, we presented a new dataset computed from testing activities performed during part of the semester at Data Structures Course. The dataset is presented and analysed in order to understand how useful it can be for predicting early student's failure. We presented several metrics for better dataset understanding and also inspected the ranking of the features using two algorithms in order to see which actions play a significant role in student's final grade at both tests and exam. Regarding this analysis, we found that the last grade obtained at the tests they took was essential and produced significant information gain.

Another focus was to evaluate how good is the dataset at predicting the tests final grade and student's failure at the final result. For this exploratory analysis, we used a selection of algorithms which covered a wide variety of situations. Regarding dataset evaluation, we obtained a good correlation for regression tasks and good accuracy for predicting the student's failure while using classification algorithms to predict the final grade was not offering good accuracy. Predicting the student's failure before the exams period is an important task, and it is useful for many learning environments as it is based on generic extracted features.

As feature work, several other algorithms are worth to be considered from both machine learning and deep learning areas. From machine learning, the gradient boosting machines are a class of algorithms that are worth exploring for better accuracy in both regression and classification tasks. Regarding deep learning algorithms, it is worth exploring neural networks that deal with numeric values. Another future work is that on the same dataset it is worth exploring other labels for predicting the tests grade or the exam's grade.

REFERENCES

1. Cortez, Paulo, and Alice Maria Gonçalves Silva. "Using data mining to predict secondary school student performance." (2008).
2. Al-Shehri, Huda, Amani Al-Qarni, Leena Al-Saati, Arwa Batoaq, Haifa Badukhen, Saleh Alrashed, Jamal Alhiyafi, and Sunday O. Olatunji. "Student performance prediction using support vector machine and k-nearest neighbor." In 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-4. IEEE, 2017.
3. Satyanarayana, Ashwin, and Mariusz Nuckowski. "Data mining using ensemble classifiers for improved prediction of student academic performance." (2016).
4. Al-Obeidat, Feras, Abdallah Tubaishat, Anna Dillon, and Babar Shah. "Analyzing students' performance using multi-criteria classification." Cluster Computing 21, no. 1 (2018): 623-632.
5. Amrieh, Elaf Abu, Thair Hamtini, and Ibrahim Aljarah. "Mining educational data to predict student's academic performance using ensemble methods." International Journal of Database Theory and Application 9, no. 8 (2016): 119-136.
6. Amrieh, Elaf Abu, Thair Hamtini, and Ibrahim Aljarah. "Preprocessing and analyzing educational data set using X-API for improving student's performance." In 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1-5. IEEE, 2015.
7. Uzel, Vahide Nida, Sultan Sevgi Turgut, and Selma Ayşe Özel. "Prediction of Students' Academic Success Using Data Mining Methods." In 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1-5. IEEE, 2018.
8. Bharara, Sanyam, Sai Sabitha, and Abhay Bansal. "Application of learning analytics using clustering data Mining for Students' disposition analysis." Education and Information Technologies 23, no. 2 (2018): 957-984.
9. Kuzilek, Jakub, Martin Hlosta, and Zdenek Zdrahal. "Open university learning analytics dataset." Scientific data 4 (2017): 170171.
10. Li, Rumei, Chuantao Yin, Xiaoyan Zhang, and Bertrand David. "Online learning style modeling for course recommendation." In Recent Developments in Intelligent Computing, Communication and Devices, pp. 1035-1042. Springer, Singapore, 2019.
11. Settles, Burr, Chris Brust, Erin Gustafson, Masato Hagiwara, and Nitin Madnani. "Second language acquisition modeling." In Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications, pp. 56-65. 2018.
12. Kormos, Judit. Speech production and second language acquisition. Routledge, 2014.
13. Koedinger, Kenneth R., Ryan SJd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper. "A data repository for the EDM community: The PSLC DataShop." Handbook of educational data mining 43 (2010): 43-56.
14. Stamper, John, Ken Koedinger, Ryan SJ d Baker, Alida Skogsholm, Brett Leber, Jim Rankin, and Sandy Demi.

"PSLC DataShop: A data analysis service for the learning science community." In International Conference on Intelligent Tutoring Systems, pp. 455-455. Springer, Berlin, Heidelberg, 2010.

15. Heffernan, Neil T., and Cristina Lindquist Heffernan. "The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching." International Journal of Artificial Intelligence in Education 24, no. 4 (2014): 470-497.
16. Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update." ACM SIGKDD explorations newsletter 11, no. 1 (2009): 10-18.

Exploring age, gender and area differences of teachers as regards mobile teaching

Gabriel Gorghiu, Elena Ancuța Santi
Valahia University Targoviste
13 Aleea Sinaia, 130004 Targoviste, Romania
ggorghiu@gmail.com, santi.anca@yahoo.ro

Costin Pribeanu
Academy of Romanian Scientists
54 Splaiul Independentei, 050094 Bucharest, Romania
costin.pribeanu@gmail.com

DOI: 10.37789/rochi.2020.1.1.25

ABSTRACT

Introducing mobile teaching and learning in schools represents a real challenge in nowadays education. Mobile devices are able to support a diversity of tasks such as communication, interaction, collaboration, information, and resource sharing. Although mobile devices are widely used for e-learning, but also very present in non-formal and informal learning, there are many barriers against the adoption of mobile technology by teachers, for teaching purposes. This paper analyzes the perceptions of Romanian teachers as regards the use of mobile technology in the teaching and learning process. The purpose of the paper is to explore the influence of gender differences, age, and area (urban vs. rural) on the perceptions of mobile teaching and learning. The results illustrate an important influence of such variables when considering the integration of mobile technology in education, as well as of a variable that has to be taken into consideration seriously: the teacher's attitude concerning the use of technology.

Keywords

Mobile devices, mobile learning, mobile teaching, group differences.

ACM Classification

D.2.2: Design tools and techniques. H5.2 User interfaces.

INTRODUCTION

Introducing mobile teaching and learning in schools is a challenge for more than a decade. Mobile devices are widely used by teachers and students for communication, interaction, socialization, collaboration, information, and resource sharing. Despite the familiarity of teachers with mobile devices, several barriers (external and internal) exist concerning the adoption of mobile technology in the teaching and learning process: technical skills needed to operate a computer, pedagogical models of technology use, teachers' personal beliefs, willingness to change, technology acceptance, lack of access, class disruption, time, training, institutional support [7, 14, 16, 18, 19].

The technology acceptance is driven by various factors, among which the most important are the perceived ease of use and the user's motivation [5]. In the context of the technology acceptance model (TAM), extrinsic motivation has been conceptualized as perceived usefulness and the intrinsic motivation has been conceptualized as perceived enjoyment [5]. Intrinsic motivation in TAM is related to user experience. Hornbaek and Hertzum [8] reviewed the relationship between technology acceptance and user experience and found that few studies are published that take into consideration the users' needs and the settings in which the technology is adopted.

Mobile devices are expected to increase students' motivation to learn and help them to better understand the lesson [13]. On another other hand, it is expected that teachers could better explain difficult concepts and better stimulate students. Understanding teachers' perception of the acceptance of mobile technology for teaching and learning also requires a closer look at demographic variables, such as gender and age [21].

Although introducing mobile devices in the teaching process creates many opportunities - as better understanding and increased motivation to learn [13, 21] -, it seems to be a difficult task, requiring an additional effort for teachers: learning how to use the new technology, how to implement, and how to design thinking by teachers [15, 16, 18, 19].

The objective of this work is to analyze the perceptions of Romanian teachers as regards the use of mobile technology in the teaching and learning process. The analysis is done on a sample of 125 teachers along with three factors: (a) expectancy of students' motivation to learn, (b) expectancy of learning usefulness, and (c) teaching usefulness. The differences as well as the perceived barriers are further analyzed by three variables: gender, age, and area (urban vs. rural).

The rest of the paper is organized as follows. In section 2, related work is discussed with a focus on the acceptance of mobile technology in schools. The method and sample are presented in section 3. Then, differences by gender, age, and area are analyzed and discussed in section 4. The paper ends with a conclusion in section 5.

RELATED WORK

The study of Mac Calum et al. [15] analyzed the drivers of teachers' acceptance of mobile teaching and learning. To do this, they extended TAM to include digital literacy, ICT anxiety, and ICT teaching self-efficacy. They found that these external variables have a significant influence on the factors that mediate the intention to use. Their study highlights the importance of digital skills needed to use mobile devices in the classroom as well as the need for support from the institution willing to promote mobile learning and teaching.

Wang et al. [21] analyzed the factors that drive the acceptance of mobile learning by using the unified theory of acceptance and use of technology [20]. They found that performance expectancy (perceived usefulness), effort expectancy (perceived ease of use), perceived playfulness (perceived enjoyment) and learning self-management are the most important drivers of acceptance. An analysis of the age differences showed that effort expectancy and social influence were stronger predictors of the intention to use the technology for the elders. Also, a gender analysis showed that social influence was a stronger predictor for men than for women, while learning self-management was a stronger predictor for women. Several studies on the issue of gender differences in the use of technology show that male subjects have a positive attitude toward technology more than females [4, 12] and they also are more confident in their abilities to use technology in learning [22], and more interested in information technology [9], although other studies found no significant relationship for age and gender, and teachers' attitudes related to exploiting the computers [17]. However, in terms of using mobile technology in learning, the studies indicate different results: several researchers confirm the existence of gender and age differences, but also social and cultural influences that can act as barriers in the implementation of m-learning [2]; while Adedoja & Morakinyo [1] show that there are no gender differences in the perceived usefulness of mobile devices in learning, in the easiness of using the means of mobile technology for learning, both categories showing a positive attitude towards m-learning.

Regarding the differences between urban and rural school teachers in the use of technology, Howley, Wood & Hough [10] show that the attitude of teachers in rural schools is positive, but those teachers seem to have less adequate skills regarding the exploitation of technology in teaching and learning process.

RESEARCH METHODOLOGY

This research is part of a larger study that started with a qualitative study aiming to understand the barriers against and motivation towards the use of mobile devices in teaching and learning [13, 16]. Based on the findings of preliminary research, an evaluation instrument has been developed, targeting several factors: motivation, learning usefulness, and teaching usefulness.

The questionnaire has been administrated during a pilot study to Romanian teachers in November-December 2019. The sample consists of 34 men and 91 women, distributed by age in groups, as follows: 15 teachers of 20-29 years, 27 teachers of 30-39 years, 29 teachers of 40-49 years, 43 teachers of 50-59 years, and 11 teachers over 60 years old. 93 teachers are working in the urban area and 32 in the rural area. First, teachers were asked to answer some general questions, then to rate several statements on a 5-points Likert scale, and last to answer two open-ended questions related to the barriers against mobile teaching and concerning the technical conditions met in their school.

The variables proposed and analyzed in this work, the mean values (M) and standard deviation (SD) are presented in Table 1.

The differences by gender, age group, and area have been analyzed by mean comparison and the one-way ANOVA test for significance.

Table 1. Variables (N=125)

Variable	M	SD
Motivation to learn		
By using mobile technology students may be *less bored* by the traditional methods	4.05	1.02
By using mobile technology students may find the lesson more *attractive*	4.25	0.89
By using mobile technology students are less stressed, and learning is accepted *as a game*	4.02	0.97
By using mobile technology students may find the lesson more *interesting*	4.34	0.86
Learning usefulness		
Mobile technology may help to learn *outside the class*	4.10	0.90
Mobile technology may help the *collaborative* learning	4.08	0.83
Mobile learning stimulates *creativity*	3.84	0.95
Mobile technology may help to better *understand* the lesson	4.08	0.79
Teaching usefulness		
With mobile technology, I could *prepare* more interesting lessons	4.26	0.80
Mobile technology helps to give learning *tasks* to students	4.06	0.79
With mobile technology, I could better *explain* difficult concepts	3.78	0.94
With mobile technology, I could better *stimulate* the students to learn	3.97	0.83

ANALYSIS OF DIFFERENCES

Differences by gender

The gender differences as regards the learning motivation, learning usefulness, and teaching usefulness are presented in Figure 1 (M=34, F=91).

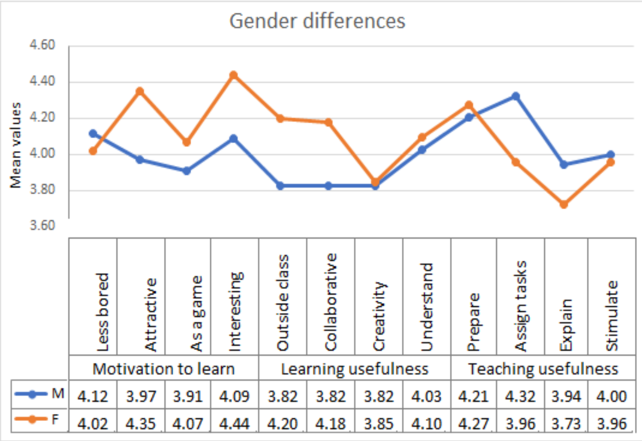


Figure 1. Gender differences

The perceptions of students' motivation expectancy and learning expectancy are higher in the case of female teachers. As regards the students' motivation to learn, the mean differences are higher and statistically significant (one-way

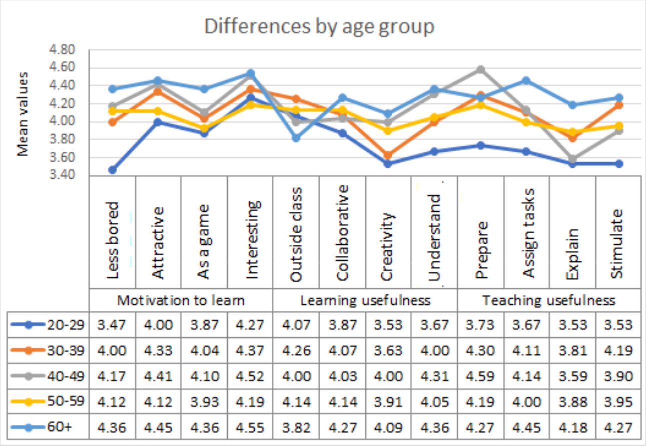
ANOVA 1, 123, 124) for the expectancy of more attractive (F=4.72, p=0.032) and more interesting lessons (F=4.42, p=0.042). As regards the expectancy of learning usefulness, female teachers also scored higher, the one-way ANOVA showing statistically significant differences for opportunities of learning outside the class (F=4.38, p=0.038) and collaborative learning (F=4.60, p=0.034).

With one exception, male teachers scored higher the items related to teaching usefulness. The one-way ANOVA showed that only one difference is statistically significant, related to assigning learning tasks to students (F=5.61, p=0.019).

As regards the barriers against the adoption of mobile technology for teaching, most frequently mentioned were the lack of equipment and/or Internet connection in school (81% of teachers) and the lack of skills (32% of male teachers and 42% of female teachers). Other mentioned barriers were: the potential of misuse of mobile devices and the lack of funds.

Differences by the age groups

The differences by the age groups are illustrated in Figure 2 (N=15/27/29/43/11).



	Less bored	Attractive	As a game	Interesting	Outside class	Collaborative	Creativity	Understand	Prepare	Assign tasks	Explain	Stimulate
	Motivation to learn				Learning usefulness				Teaching usefulness			
20-29	3.47	4.00	3.87	4.27	4.07	3.87	3.53	3.67	3.73	3.67	3.53	3.53
30-39	4.00	4.33	4.04	4.37	4.26	4.07	3.63	4.00	4.30	4.11	3.81	4.19
40-49	4.17	4.41	4.10	4.52	4.00	4.03	4.00	4.31	4.59	4.14	3.59	3.90
50-59	4.12	4.12	3.93	4.19	4.14	4.14	3.91	4.05	4.19	4.00	3.88	3.95
60+	4.36	4.45	4.36	4.55	3.82	4.27	4.09	4.36	4.27	4.45	4.18	4.27

Figure 2. Age group differences

Overall, younger teachers (20-29 years) seem to have the lowest confidence in mobile teaching and learning. On the contrary, teachers over 60 years old have the highest perceptions with one exception.

A one-way ANOVA (4, 120, 124) shows a statistically significant difference for the opportunity to prepare more interesting lessons (F=3.12, p=0.02) and a marginal significance for better understanding expectancy (F=2.17, p=0.076).

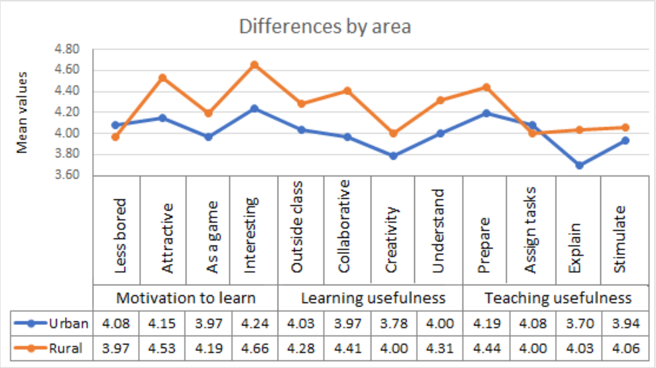
The lack of skills needed to adopt mobile teaching has been mentioned mainly by older teachers, respectively 64% of teachers over 60 years old, 48% of teachers having 40-49 years old, and 37% of teachers having 50-59 years old.

Differences by area

The differences by the area where the school is located are presented in Figure 3 (urban=93, N=26).

Overall, teachers working in rural areas have a higher perception related to the learning motivation and usefulness brought by the presence of mobile technology.

A one-way ANOVA (1, 123 124) shows that the differences are statistically significant for the following variables: the expectancy of attractive lessons (F=4.52, p=0.035), more interesting lessons (F=5.86, p=0.017), and collaborative learning (F=6.98, p=0.009). The differences are marginally significant for better understanding expectancy (F=3.82, p=0.053) and the opportunity to better explain difficult concepts (F=3.03, p=0.084).



	Less bored	Attractive	As a game	Interesting	Outside class	Collaborative	Creativity	Understand	Prepare	Assign tasks	Explain	Stimulate
	Motivation to learn				Learning usefulness				Teaching usefulness			
Urban	4.08	4.15	3.97	4.24	4.03	3.97	3.78	4.00	4.19	4.08	3.70	3.94
Rural	3.97	4.53	4.19	4.66	4.28	4.41	4.00	4.31	4.44	4.00	4.03	4.06

Figure 3. Area differences

As regards the barriers, the lack of equipment and/or Internet connection was mentioned by 75% of teachers working in the urban area and 85% of teachers working in a rural area, while the lack of skills was mentioned by 39% of teachers working in the urban area and 38% of teachers working in rural areas.

DISCUSSION

The results of the study show that there are gender differences concerning the teachers' perception of the role of mobile technology in teaching and learning. Female teachers consider more than males that mobile technology represents a motivational variable for students (lessons become more attractive, more interesting) and useful in learning (create opportunities of learning outside the class and collaborative learning). Male teachers appreciate that mobile technology is perfect for assigning learning tasks to students, being also useful for explaining theoretical notions.

The results are similar to the findings of researches, who identified gender as an important variable in the way that teachers perceive and use technology in education. We appreciate those differences as related to intrinsic factors - mainly with the teaching style and personal vision on education, but also cultural and social factors (according to [2]). Also, the results illustrate the age as a factor that influences mobile teaching. Interestingly, young teachers do not benefit in a great manner of such opportunities - a possible explanation could be that beginners are usually focused on getting teaching positions in rural schools, but the schools' facilities and also the students' financial possibilities are limited on using modern technology in the educational process (lack of equipment and/or Internet connection, in many cases). Anyway, it is important to notice that teachers of 60+ recorded good scores, but the 50-59 age group and the 40-49 years old teachers show more availability for using mobile technology in teaching and learning, being groups with rich experience, possibilities and facilities in schools, also with a great interest for the career (according to with the psychosocial development stages of Erikson's theory).

Similar to Howley, Wood & Hough [10], the results indicate that teachers from rural schools have a higher perception concerning the increase of learning motivation and usefulness of mobile teaching, although in many cases, they face much more difficulties than in schools from the urban areas.

By sure, there are limitations of this exploratory study. First, the sample of the research is not very extensive (125 subjects), so that the results cannot be generalized at the national level. Secondly, the distribution by gender and area is not very balanced: 34 men and 91 women, with 93 teachers working in the urban area and 32 in the rural area. Another limitation is coming from the research itself: it is based, at this level, only on the teacher's perceptions and answers - the student's perspective or other evaluation issues are missing, so that it may induce a subjective factor.

CONCLUSION AND FUTURE WORK

This study contributes to a better understanding of the factors that drive the adoption of mobile technology for teaching and learning. The results show that a series of variables can influence the process of integrating modern technology into education (age, gender, local area), but also the attitude of teachers remains important: teachers who value such resources as necessary and useful improve their teaching knowledge and can build meaningful learning experiences for students. In the actual society, technology has become indispensable and teachers must have not just specific equipment at their disposal, but also necessary digital skills and abilities, as well as a desire and ambition for changing the educational process, by considering the student in its central point.

REFERENCES

1. Adedoja, G., Morakinyo, D. A. (2016). Gender Influence on Undergraduates Students' Acceptance of Mobile Learning Instruction using Technology Acceptance Model (TAM). *Asian Journal of Education and e-Learning*, *4*(2), 65-70.
2. Al-Hunaiyyan, A., Alhajri, R., Al-Sharhan, S. (2017). Instructors Age and Gender Differences in the Acceptance of Mobile Learning. *International Journal of Interactive Mobile Technologies*, *11*(4), 4-15. DOI: 10.3991/ijim.v11i4.6185
3. Chiu, T., Churchill, D. (2016) Adoption of mobile devices in teaching: changes in teacher beliefs, attitudes, and anxiety, *Interactive Learning Environments*, *24*(2), 317-327, DOI: 10.1080/10494820.2015.1113709
4. Comber, C., Colley, A. (1997). The effects of age, gender and computer experience upon computer attitudes. *Educational Research*, *39*(2), 123-133.
5. Davis, F. D., Bagozzi, R. P., Warshaw, P. R. (1992). Extrinsic and intrinsic motivation to use computers in the workplace. *Journal of Applied Social Psychology*, *22*(14), 1111-1132.
6. Erikson, E.H. (1982) The life cycle completed. New York: Norton.
7. Ertmer, P. A. (1999). Addressing first-and second-order barriers to change: Strategies for technology integration. *Educational technology research and development*, *47*(4), 47-61.
8. Hornbæk, K., Hertzum, M. (2017). Technology Acceptance and User Experience: A Review of the Experiential Component in HCI. *ACMTrans. Comput.-Hum. Interact.* *24*(5), Article 33 (October 2017), 30 pages, DOI: 10.1145/3127358.
9. Houtz, L. E., Gupta, U. G. (2001). Nebraska high school students' computer skills and attitudes. *Journal of Research on Computing in Education 33*(3) 316-328.
10. Howley, A., Wood, L., & Hough, B. (2011). Rural elementary school teachers' technology integration. *Journal of Research in Rural Education*, *26*(9), 1-13.
11. Huang, J. H., Lin, Y. R., Chuang, S. T. (2007). Elucidating user behavior of mobile learning. *The electronic library*, *25*(5), 585-598.
12. Kadijevich, D. (2000). Gender differences in computer attitude among ninth-grade students. *Journal of Educational Computing Research*, *22*(2), 145-154.
13. Lamanauskas, V., Slekiene, V., Gorghiu, G., Pribeanu, C. (2019). Better learning and increased motivation to learn with mobile technology (devices): A preliminary study. *Natural Science Education 16*(2), 80-88.
14. Leem, J., Sung, E. (2019). Teachers' beliefs and technology acceptance concerning smart mobile devices for SMART education in South Korea. *British Journal of Educational Technology*, *50*(2), 601-613. https://doi.org/10.1111/bjet.12612
15. Mac Callum, K., Jeffrey, L., Kinshuk. (2014). Factors impacting teachers' adoption of mobile learning. *Journal of Information Technology Education: Research*, *13*, 141-162.
16. Pribeanu, C., Gorghiu, G., Lamanauskas, V., Slekiene, V. (2020) Use of mobile technology in the teaching/learning process: opportunities and barriers. *Proceedings of ELSE 2020 Conference*, Vol I, 376-383, DOI: 0.12753/2066-026X-20-049.
17. Teo, T. (2008). Pre-service teachers' attitudes towards computer use: A Singapore survey *Australasian Journal of Educational Technology*, *24*(4), 413-424, DOI: 10.14742/ajet.1201.
18. Tsai, C. C., Chai, C. S. (2012). The "third"-order barrier for technology-integration instruction: Implications for teacher education. *Australasian Journal of Educational Technology*, *28*(6), 1057-1060.
19. Thomas, K., O'Bannon, B., Bolton, N. (2013). Cell Phones in the Classroom: Teachers' Perspectives of Inclusion, Benefits, and Barriers, *Computers in the Schools*, *30*(4), 295-308.
20. Venkatesh, V., Morris, M. G., Davis, G. B. Davis, F. D. (2003). User acceptance of information technology: toward a unified view. *MIS Quarterly*, *27*(3), 425-478.
21. Wang, Y. S., Wu, M. C., Wang, H. Y. (2009). Investigating the determinants and age and gender differences in the acceptance of mobile learning. *British Journal of Educational Technology*, *40*(1), 92-118.
22. Yau, H. K., Cheng, A. L. F. (2012). Gender Difference of Confidence in Using Tehnology for Learning. *Journal of Technology Studies*, *38*(2), 74-79.