SSLD:
a French SMS to Standard Language Dictionary

Louise-Amélie Cougnon[1], Richard Beaufort[1]
Université catholique de Louvain - CENTAL

Abstract

This paper presents a methodology to semi-automatically build up a dictionary out of an SMS corpus. First, we describe the three-step approach that extracts the dictionary entries from the corpus and we detail the smart manual sorting performed on the dictionary. Then, we give a panorama of SMS phenomena observed in the dictionary. Finally, we survey the current limits of our methodology and the related improvements that can be made to it.

Keywords: SMS dictionary, SMS phenomena, alignment, finite-state machines.

1. Introduction

At a time when technology intensifies and strengthens mechanical and human communication over the world, the study of new types of dictionaries and lexical resources seems essential. The language used in Short Message Service (SMS), on the same level as *chat* language, is one of these new written forms of communication. When dealing with SMS[2], one has to cope with various issues: new linguistic phenomena, language processing difficulties and lexical resource limits. Linguistic phenomena in SMS go from phonetic and numeral scripts, abbreviations and capital letters, to intensive use of neologisms, language mixing and borrowing, through new code systems such as emoticons. Processing SMS corpora involves identifying lexemes, applying dictionaries and using particular tools such as taggers, grammatical analyzers and lexical resources. There is a wide range of lexical resources for SMS studies, but unfortunately, studies on transcription from SMS to standard language are few and results are still too basic (mainly because they are based on corpora of limited size[3]).

The *sms4science* project aims at collecting international SMS corpora. Since the beginning of the project, we have been questioning the usefulness of SMS corpora and

[1] {louise-amelie.cougnon,richard.beaufort}@uclouvain.be

[2] The acronym refers to the service as much as to the messages exchanged during the service.

[3] Guimier de Neef and Fessard (2007) is a notable exception as they made use of a corpus of 10,000 SMS.

SMS to standard language transcription. At this stage, we had already worked on SMS transcription, especially on the reverse dictionary, viz. standard to SMS language (http://www.uclouvain.be/44009.html). Then, in early 2008, a new project was set up within the framework of our research centre: *Vocalise*, an SMS-to-Speech synthesis project (http://cental.fltr.ucl.ac.be/projects/vocalise/index_EN.html). In order to improve SMS speech synthesis, the new project developed an SMS word alignment system based on a corpus of 30,000 text messages and their manual transcription[4]. It was, for us, the opportunity to address the question of SMS to standard language transcription again. We decided to use the *Vocalise* aligned corpora to draw up an SMS to Standard Language Dictionary (SSLD). In order to meet this objective, we built a list of entries based on all the words of the aligned corpora.

This paper is organized as follows. Section 2 presents the three-step approach, which made it possible to semi-automatically build up a dictionary out of an SMS corpus, while Section 3 focuses on the smart manual sorting of the dictionary and presents the SMS phenomena (which we refer to as "categories") that constitute the dictionary entries. Section 4 details the kinds of mistakes our three-step approach made at different levels, and proposes some possible improvements, which should significantly enhance the SSLD-making procedure. We finally draw some conclusions in Section 5.

2. From SMS-gathering to dictionary-making

Three distinct steps enabled the dictionary making: corpus collection and transcription (part of the sms4science project), corpus alignment (part of the Vocalise project) and raw SMS resource extraction.

2.1. Corpus collection and transcription

We built up the SSL dictionary from a French SMS corpus of 30,000 messages, gathered in Belgium, semi-automatically anonymized and manually normalized[5] at the Université catholique de Louvain (Fairon and Paumier, 2006). As shown in Figure 1, the SMS corpus and its transcription constitute parallel corpora aligned at the message-level.

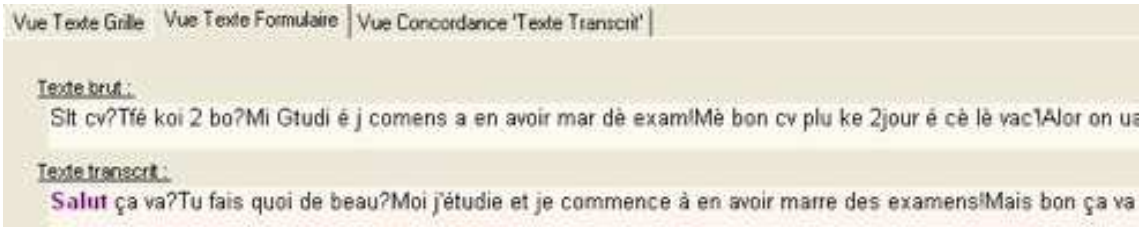

Figure 1. Snapshot of message-level aligned corpora

[4] The project "Faites don de vos SMS à la science" collected 30,000 French text messages in 2004.

[5] *"SMS normalization consists in rewriting an SMS text using a more conventional spelling, in order to make it more readable for a human or for a machine"* (Yvon 2008).

2.2. Corpus alignment

Unfortunately, this message-level alignment does not allow for pertinent lexical extraction and equivalence. In order to achieve this purpose, we needed an alignment at the word level: for each word of a sentence in the standard transcription, we had to know the corresponding sequence of characters in the SMS version. As an accurate automatic linguistic analysis of the SMS corpus was not possible, we needed another way of producing this word-alignment: a method able to align sentences at the character level. This method is called "string alignment"[6]. One way of implementing this string alignment is to compute the edit-distance of two strings, which measures the minimum number of operations (substitutions, insertions, deletions) required to transform one string into the other (Levenshtein, 1966). Using this algorithm, in which each operation gets a cost of 1, two strings may be aligned in different ways with the same global cost. For instance, the couple (*kozer*, *causé*) could be aligned:

```
(1) ko_ser      (2) k_oser      (3) ko_ser      (4) k_oser
    causé_          causé_          caus_é          caus_é
```

where underscores (_) mean "insertion" in the upper string, and "deletion" in the lower string. However, from a linguistic standpoint, only alignment (1) is desirable, because corresponding graphemes are aligned on their first character. In order to automatically choose this preferred alignment, we had to distinguish the three edit-operations, according to the characters to be aligned. For that purpose, probabilities were required. Computing probabilities for each operation according to the characters to be aligned was performed through the following iterative algorithm, implemented in the framework of the Vocalise project:

STEP 1. Align the corpora using the standard edit-distance (with edit-cost of 1).

STEP 2. From the alignment, learn probabilities of applying a given operation on a given character.

STEP 3. Re-align the corpora using a weighted edit-distance, where the cost of 1 is replaced by the probabilities learned in STEP 2.

STEP 4. If two successive alignments provided the same result, there is a convergence and the algorithm ends. Else, it goes back to STEP 2.

Hence, the algorithm gradually learns the best way of aligning strings. On our SMS parallel corpora, the algorithm converged after seven iterations and provided us with a result (see Figure 2) from which the lexicon of SMS words could be built.

A standard way of implementing edit-distance is to use dynamic programming (Viterbi, 1967). However, in order to easily compute weighted edit-distances, we used weighted finite-state machines, which were shown by Mohri (2003) to be very

[6] String alignment comes from bioinformatics, where sequences of DNA must be arranged in such a way that similarities and differences are identifiable.

efficient in this task. The finite-state library in use here is described in Beaufort (2008), and the finite-state alignment of the iterative algorithm (steps 1 and 3) is detailed in Beaufort *et al.* (2008).

```
28620: S_t_t c__v_?_T__fé__ k_oi 2_ b_o_?_M_i G__tudi_ é_ j_ com_ens_ a
28620: Salut ça va? Tu fais quoi de beau? Moi j'étudie et je commence à
```

Figure 2. Snapshot of the word-level alignment corpora, where symbols _ stand for insertions and deletions

2.3. SSLD input extraction

Based on this character-level alignment, an extraction script[7] enabled us to extract, for each sequence, its raw and standard variants. The script loaded a regular French language dictionary[8] that allowed matching our SMS standard sequences with recognised inflected forms and their lemma. In our SSLD, each entry is not followed by its standard sequence, but by its lemma, as can be seen in Figure 3.

```
monitric | ~~(monitrice)~~ | moniteur | N+z1:fs
```

Figure 3. SSLD extract showing the unwanted (standardised inflected) column

For ambiguous sequences that showed various lemmas, a new entry was created for each possible grammatical interpretation. Figure 3 also shows that the SMS sequences and the lemma are followed by their grammatical and inflectional information and potentially, by additional information, such as lexical layers (z1, z2, z3, etc.) and semantic information (as *Hum* for any name referring to a person or *Profession* for any name referring to a profession, etc.)[9].

The extraction script mainly implements the following algorithm:

```
STEP 1. For each aligned pair {SMS message, standard message},
        Split the two messages according to blanks and punctuations
         in the standard message
        For each pair of {SMS, standard} segments
        Clean segments (remove insertion and deletion symbols _,
         convert each upper case into the corresponding lower case)
        Store the pair in a temporary lexicon, except if the SMS
         sequence is empty or matches with a number/time pattern
STEP 2. For each stored pair from the temporary lexicon,
        If the standard word exists in the DELAF lexicon,
         for each DELAF lexicon entry {standard word, lemma,
         category}, create a new SSLD entry {SMS sequence, lemma,
         category}
        Else, create a new SSLD entry, {SMS sequence, UNKNOWN tag}
```

[7] Our gratitude goes to Hubert Naets, who wrote this script.

[8] The DELAF was our reference dictionary; it is an electronic dictionary for French, initially built up by M. Gross and mainly developed during the 80's and 90's. It includes 683,824 entries for 102,073 different lemmas.

[9] Our system of codes is totally inspired by Unitex dictionaries syntax.

After the application of this script on our aligned corpora, the SSLD lexicon comprised 45,049 entries for 10,318 different lemmas[10].

3. Smart sorting and analysis of the SSLD

3.1. Smart sorting

At this step, we manually filtered out unwanted entries so as to obtain a smarter SSLD. All unknown sequences added to the SSLD by the extraction script were manually revised: neologisms (later than 2001[11]), word plays, proper names (toponyms, first names and trade marks), foreign words (*monkey*, *besos*, *aanwezig*, etc.), unrecognised sign/number patterns (e.g. *07h5* for *07h05*), emotive graphics (e.g. repetition of letters showing intensity) and transcriber's mistakes (*cnpine* for *copine* 'girl friend', *premdr* for *prendre* 'to take', etc.). All these categories were kept, apart from proper names and transcriber's mistakes.

During this checking task, each SSLD entry was also labelled with one of the seven SMS categories (presented in section 3.2) we defined in order to characterize the stylistic phenomena of the SMS corpus. Some ambiguous sequences, however, could not be directly associated with any of our categories, and we had to go back to the initial corpus and look at the context. For instance, the entry *rè*, whose lemma was *trait*, was difficult to label: we clearly could have thought of an abbreviating phenomenon (added to some sort of phonetisation), while *rè* was just the last segment of the SMS form *pRmeterè*, which stood for *permettrait* ('would permit') and had been wrongly segmented into 2 entries by the extraction script.

3.2. Analysis

3.2.1. Seven SMS categories

First of all, and contrary to what one might think, standard inflected words that satisfy standard spelling make up half of our SSLD entries. On the other half of the entries, some SMS phenomena were rapidly recognized: the abbreviating process is commonly known, as well as phonetisation (which is a subcategory of abbreviation), which describes letters, numbers or signs used for their phonetic values[12]. We chose to distinguish the use of signs and the use of numbers. We finally added the "mistakes" category (which includes SMS user, transcriber, word-aligner or algorithm mistakes) and the "unlikelies" category, which are not SMS phenomena strictly speaking but which have to be considered apart from other SMS phenomena. None of these

[10] Our gratitude goes to Master students in philology who helped us sorting out the dictionary entries.

[11] Unfortunately, the DELAF dictionary has not been significantly upgraded since 2001.

[12] A letter used for its phonetic value is spelt, instead of being simply pronounced like in word context.

categories was deleted as they all conveyed specific information that could be used to improve automatic SMS reading and understanding.

The phonetisation category had to be specified. Since we decided to put numbers and signs aside, this category was used to define any sequence that phonetically resembled the standard word. We put in this category phonetisation strictly speaking (e.g. *pnible* for *pénible*), any sequence showing schwa deletion (e.g. *bêtis* for *bêtise*), but also any simplification that maintains the phonetic resemblance (e.g. *ail* for *aille*, the subjunctive of *aller*, "to go"). This category is by far the most popular SMS graphic phenomenon, because it includes any unaccentuated word.

3.2.2. The "unlikelies"

The fact that, for ambiguous terms, a new entry is created for each possible lemma, ensures a certain improvement of the dictionary; but it also adds some ambiguity if, for example, the SSLD was to be used for automatic translation. For terms which could be either nouns or inflected verbs (e.g. *échange*), the ambiguity has to be maintained and could probably be solved by the context. But in other cases, the confusion is unnecessary, because one of the lemmas is very frequent, while the others are fairly rare, at least in SMS context. This is what we called an "unlikely": a rare lemma. All unlikelies were deleted from the dictionary.

Example	**Meaning**
`ballons,baller.V:P1p:Y1p`	"to dance, to jolt"
`muchas,mucher.V:J2s`	"to hide"

Figure 4. Examples of unlikelies in the SSLD

The second example of Figure 4 is of a particular interest: the French homograph of this Spanish word is not frequent enough to maintain an entry in the SSLD dictionary. Nevertheless, we decided not to delete this kind of entries, but to mark them with a special *unlikely* tag that would allow us to identify and delete them later.

3.2.3. Unknown sequences

As we reported above, a sizeable part of unknown words that we reintroduced in the dictionary were words that entered the French language after 2001. These words mostly refer to new realities (*fitness*, *monoparentalité*), or technologies (*adsl*, *bipeur*[13], *pirater*). Some of them, however, are just new labels for well-known realities (*criser* "to be on edge", *tilter* "to suddenly understand", *cafariser* "to sadden", or *moisversaire* "a celebration that happens the same day of each month").

[13] The verb *biper* can be found in the DELAF but not the noun *biper* and its alternative spelling *bipeur*.

Some other sequences labelled as unknown turned out to belong to some specific terminology: *acerifolia* (botany), *markopsien* (marketing) and *émollience* (cosmetics) are good examples of this phenomenon. We decided to keep them as part of the SMS user's lexicon.

Finally, a lot of unknown entries were identified as regionalisms, and included in our final dictionary. As our corpus was collected in Belgium, regionalisms were mostly Belgian or at least shared by Belgium and other French-speaking areas. Words like *baraki*, *berdeller*, *copion*, *guindaille* and *se faire carotter* illustrate this clear trend.

4. Issues and possible improvements

This section presents the different kinds of mistakes that occurred at various stages of our methodology, and the possible solutions we propose to solve them.

4.1. Manual transcription

As a matter of fact, first mistakes are due to the transcriber himself. Even when he carefully checks his work, a single transcriber is not enough to avoid accidental mistakes, which of course occurred quite frequently for a 30,000 SMS corpus. Naturally, we could help the transcriber by checking his transcription several times. However, to err is human, and even multiple checking will not point out all mistakes. A complementary solution could be to automatically perform lexicon looks-up during the transcription process, and to draw the transcriber's attention to possible out-of-vocabulary words or infrequent forms.

4.2. Alignment algorithm

Three kinds of mistakes are due to the alignment algorithm. First, cases of agglutination are frequent: the aligner shows a clear tendency to align on the first of two words when a letter is missing (cf. Figure 5). Second, some typography is not handled, such as the *&* symbol not recognized as *et*, or the digit *1* identified as being the letter *i* (cf. Figure 6). Third, some subtle cases of phonetisation are not taken into account by the process. This is the case with letters, numbers or signs that replace more than one word.

```
D__t__t__Facon_J_en_Ai Plu__Besoin:-D D__c_Fo___PluS__tréssé_..
De toute façon j'en ai plus besoin:-D Donc faut plus stresser..
```

Figure 5. False alignments

```
G____besoin_2__partaG__ k___k1_s____tan__ a__c toi
J'ai besoin de partager quelques instants avec toi
```

Figure 6. Typography problems

These errors are due to the fact that the alignment works without resort to linguistics; it simply iteratively computes affinities of association between *letters*, and uses them to gradually improve the character-level alignment. However, as recent linguistic studies showed, phonetic transcriptions (*sré* instead of *serai*, '[I] will be', *kom* instead of *comme*, 'as') and phonetic plays (*2m1* instead of *demain*, 'tomorrow', *k7* instead of *cassette*, 'tape') are very frequent in SMS. This could be exploited by the alignment, which could perform its task *through* a phonetic version of the sequences to be aligned. Figure 7 gives an example of phonetic alignment that solves a kind of error depicted in Figure 6.

```
      SMS text:               k _ _ _ k _ _ _  _  1 _ s t a n _ _
      SMS phonetisation:      k _ _ _ k _ _ _  _  e~_ s t a~_ _ _
      Standard phonetisation: k _ E l k _ @ z  _  e~_ s t a~_ _ _
      Standard text:          q u e l q u e s ' ' i n s t a n t s
```

Figure 7. Phonetic alignment. The phonetic alphabet in use is SAMPA.

Of course, here, an important fact must be taken into account: while a standard written sentence can be automatically analyzed and unambiguously phonetised by NLP applications, it is not the case for an SMS sentence, which is difficult to analyze, and should thus be transcribed as a lattice of possible phonetisations. The alignment, here, will thus face another problem: the weight of these concurrent phonetisations, in order to choose the best path in all possible phonetic alignments.

4.3. Extraction algorithm

The extraction algorithm also showed some limits. First issues are due to the deletion of characters considered as separators: some ambiguous characters considered as separators were lost, while they were used as signs for phonetic purposes or abbreviation (cf. Figure 8). However, keeping extra punctuation would have generated too much noise.

```
      Ben viens-chercher-la' clé usb au -sud. 18-à tout de suite.
      Ben viens-chercher-la_ clé USB au _Sud_ 18-À tout de suite.
>>
      -sud,sud.N+z1:ms
      -sud,sud.A+z1:ms:fs:mp:fp
```

Figure 8. Punctuation mismatch

The second loss of information is due to the systematic neutralization of the case, as most upper-case characters were at the beginning of sentences. Nevertheless, some upper case letters carried pieces of phonetic information that would have been useful in the reading of dictionary entries (e.g. the *T* in *arT* for *arrête* is always upper case).

The third problem related to identical buffers void of letters or numbers. While it was needed to delete any number or time expression from our dictionary, it was also

unfortunate to lose all character sequences that could have carried information (e.g. emoticons).

Actually, all these limitations have a single origin: the extraction algorithm rates a couple of aligned sentences just as two strings of characters, and makes arbitrary choices only based on predefined sets of characters (letters, punctuations, symbols, etc.), without taking the context into account. Based on this observation, we consider the possibility to provide the algorithm with an automatic morphosyntactic analysis of the normalized side of the alignment. This linguistic analysis should help the algorithm split the sentence into the right segments, and add the right entries to the SSLD.

4.4. False entries

Plays on letters were hardly dealt with by the system, because even when both the alignment and the extraction steps did not generate errors, some sequences did not correspond to lexical entries and should have been left out of the dictionary (cf. Figure 9). Just as the extraction algorithm, false entries could be rejected by the system, by checking their linguistic analysis through an automatic analyzer.

```
7_____rop b_o_ 7____ idylle k_i 7__ternise
C'est trop beau cette idylle qui s'éternise
```

Figure 9. False entries

5. Conclusions and prospects

The dictionary built up using this framework is not exhaustive at all: it covers neither all lexical fields, nor the whole lexicon of any particular field. However, it gets credit for covering an important part of Belgian SMS users' lexicon. It might thus be of interest to have it further examined and compared to standard dictionaries. Which proportion of a standard dictionary is really covered? Are there new spellings and new words in this dictionary – and not in standard ones – which should be included in them? Sequences like *asap*, *lol*, *mdr* (*mort de rire*, stands for *lol* in French), *admin*, are commonly understandable and are not local, or part of a specific terminology or register. Could a standard dictionary be inspired by our SSLD which is based on genuine written practices?

Some improvements of our dictionary might be beneficial. At first, our methodology will be improved, starting with a special focus on the problems raised in Section 4. But we will also enhance our lexicon, by applying our methodology to three new significant French-speaking corpora, gathered in Québec (2009), Switzerland (2009) and France (2010). Finally, we will improve our results by applying a more recent electronic dictionary. In order to improve the dictionary, further studies could also focus on the use of cases (are capital letters always phonetisations?) and specific

characters (is the absence of schwa, like in the opposition *échang* – noun – and *échange* – verb –, significant for grammatical desambiguation?).

Finally, the SSLD is not only a starting point for linguistic studies. This resource is also fundamental for SMS-based applications, like text-to-speech synthesis applied to SMS messages. The automatic linguistic analyzer included in any speech synthesiser uses lexical resources to both disambiguate and phonetise the words that must be read aloud by the system. Faced to SMS messages with noisy forms, an analysis that only relies on standard lexica will fail, while a specialized dictionary like the SSLD should make it easier to find out the standard written word hidden behind a given noisy form. In this context, a reliable SSLD should thus be a real improvement. The Vocalise project, which provided us with the alignment algorithm, is based on this assumption.

References

BEAUFORT, R. (2008). *Application des machines à états finis en synthèse de la parole. Sélection d'unités non uniformes et correction orthographique*. PhD Thesis. Department of Semantics and Computational Logic, Faculty of Computer Science, FUNDP, Namur, March 4th 2008.

BEAUFORT, R., ROEKHAUT, S. and FAIRON, C. (2008). Définition d'un système d'alignement SMS/français standard à l'aide d'un filtre de composition. In *Proceedings of the 9th International Conference on the Statistical Analysis of Textual Data (JADT 2008)*. Lyon. March 12-14, 2008.

COUGNON, L.-A. (forthcoming). La néologie dans 'l'écrit spontané'. Etude d'un corpus de SMS en Belgique francophone. In *Actes du Congrès International de la néologie dans les langues romanes*. Barcelone. 7-10 mai 2008.

FAIRON, C., KLEIN J.R. and PAUMIER S. (2006a). *Le langage SMS*. Louvain-la-Neuve, Presses universitaires de Louvain (*Cahiers du Cental*, 3.1).

GUIMIER DE NEEF, E. and FESSARD, S. (2007). Évaluation d'un système de transcription de SMS. In *Actes du 26e Colloque international Lexique Grammaire*, Bonifacio, 2-6 octobre 2007: 217-224.

LEVENSHTEIN, V. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics*, 10: 707-710.

MOHRI, M. (2003). Edit-distance of weighted automata: General definitions and algorithms. *International Journal of Foundations of Computer Science*, 14(6): 957-982.

PANCKHURST R. (2008). Short Message Service (SMS) : typologie et problématiques futures. Arnavielle T. (coord.), *Polyphonies*, pour Michelle Lanvin, Montpellier, Éditions LU: 33-52.

VITERBI, A.-J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2): 260-269.