Normalizing SMS: are two metaphors better than one ?

Catherine Kobus
Orange Labs
2, avenue Pierre Marzin
F-22300 Lannion

François Yvon
Univ Paris-Sud 11 & LIMSI/CNRS
BP 133
F-91403 Orsay Cedex

Géraldine Damnati
Orange Labs
2, avenue Pierre Marzin
F-22300 Lannion

{catherine.kobus,geraldine.damnati}@orange-ftgroup.com, yvon@limsi.fr

Abstract

Electronic written texts used in computer-mediated interactions (e-mails, blogs, chats, etc) present major deviations from the norm of the language. This paper presents an comparative study of systems aiming at normalizing the orthography of French SMS messages: after discussing the linguistic peculiarities of these messages, and possible approaches to their automatic normalization, we present, evaluate and contrast two systems, one drawing inspiration from the Machine Translation task; the other using techniques that are commonly used in automatic speech recognition devices. Combining both approaches, our best normalization system achieves about 11% Word Error Rate on a test set of about 3000 unseen messages.

1 Introduction

The rapid dissemination of electronic communication devices (e-mails, Short Messaging Systems (SMS), chatrooms, instant messaging programs, blogs, etc) has triggered the emergence of new forms of written texts (see eg. (Crystal, 2001; Véronis and Guimier de Neef, 2006)). Addressed to relatives or peers, written on the spur of the moment, using interfaces, each with its specific constraints (computer keyboards, PDAs, mobile phones keypads), these electronic messages are characterised by massive and systematic deviations from the orthographic norm, as well as by a non conventional use of alphabetical symbols. In fact, letter and punctuation marks are not only used to conventionally encode a phonetic content, but also to introduce meta-discourse, and to signal emotions, verbal effects (eg. laughters), or attitudes (humor, derision, emphasis etc). If each media enforces its own set of constraints and promotes idiosyncratic forms of writings, these new types of texts nonetheless share a lot of commonalities. To effectively process these messages, it is thus necessary to develop robust language processing tools, capable of bearing with the extreme form of "noise" they contain. In this study, we focus more specifically on SMS, which, due to the paucity of their input interface (mobile phone keypads) seem to constitute the most challenging type of data. To a large extent, the techniques we present in this paper are also applicable to other types of electronic messages.

The "SMS language" (or "texting language") has been the subject of several linguistic studies (notably, for French, (Anis, 2001; Fairon et al., 2006)), which have emphasized its main characteristics, amongst which the extraordinary orthographic variability of lexical forms. In brief, this variability results partly from the mixing of several encoding systems: in SMS, the usual alphabetic system competes with a more "phonetic" type of writing (e.g. *rite* for *right*[1]), as well as with traces of a "consonantic" spelling (vowels are deleted, as in *wrk* for *work* or *cn* for *can*), and with non-conventional use of letters or numbers, sometimes used to encode the phonetic value of their

[1]We will illustrate this general presentation of the SMS language using examples taken from English messages, even though our systems deal with French messages. As far as we can see, the same types of deviations from the orthographic norm are observed in both languages, albeit in different proportions. A more thorough comparison of both languages certainly remains to be carried out.

spelling, as in *ani1* for *anyone*. These spelling systems can also be mixed as in *Rtst* (for *artist*) or *bcum* (for *become*). This variability is also the result of an informal style of communication, which licenses many deviations from the orthographic (simplification of repeated consonants, use of non-conventional abreviations) and grammatical (absence of case distinction, erratic use of punctuation marks, non-respect of agreement or tense markers, etc) prescriptions, notwithstanding truly unintentional typos. Finally, practitioners of the texting language excel in devising acronyms which condense, sometimes in a radical way, multi-word units: this is for instance the case with *afair*, which stands for *as far as I recall*. As a result, from a natural language processing (NLP) point of view, these messages contain an abnormally high rate of out-of-vocabulary forms, and the ambiguity of existing word forms is aggravated, two factors that contribute to degrade the performance of natural language processing tools. Recovering a normalized orthography seems thus to be a necessary preprocessing step for many real-world NLP applications, such as text-to-speech, translation, or text mining applications (filtering, routing, information retrieval, etc).

These short messages have so far received relatively little attention from the NLP community[2]: see, for English, (Aw et al., 2006; Choudhury et al., 2007), which both address the problem with statistical learning techniques, and, for French, (Guimier de Neef et al., 2007), which details a complete pipe-line of hand-crafted, symbolic, modules. In fact, the problem of normalizing SMS shares a lot of commonalities with other NLP applications, and can be addressed from several viewpoints. The first, maybe the most natural angle, is to make an analogy with the spelling correction problem. This problem has been extensively studied in the past and a variety of statistical approaches are readily available, most notably the “noisy channel” approach (see eg. (Church and Gale, 1991; Brill and Moore, 2000; Toutanova and Moore, 2002)). An alternative metaphor is the translation metaphor: under this view, the normalization task is accomplished by taking the SMS language as a foreign language, and using standard (statistical) translation techniques. Both views have their own merit, and their limitations, which we shall review shortly. In this paper, we propose yet another metaphor, which stems from the similarities between the SMS language and speech, and notably from the fact that in SMS, word separators are much less reliable than in conventional writings. As a result, it seems necessary to implement techniques, which are, as in speech recognition, capable of recovering the correct word segmentation.

The main contribution of this work is to present, evaluate and contrast two approaches to the SMS normalization problem. We demonstrate that both approaches have different advantages and pitfalls, and show their combination yields significant improvements wrt. to both single systems.

This paper is organized as follows: in section 2, we discuss these three metaphors; then go on to describe our own implementation of two different systems for normalizing SMS in 3. A comparative evaluation of these systems is conducted in section 4, where we emphasize the complementarity of both approaches, and assess the performance of a combined systems. Section 5 presents a summary of the main findings and future prospects.

2 Three metaphors for SMS normalization systems

In this section, we set the general problem of SMS normalization, and discuss, based on the analysis of SMS examples, the relevance of various NLP approaches to this task.

2.1 The "spell checking" metaphor

A first approach to the problem considers each input token as “noisy” version of the correct word form: normalization is thus viewed as a spell checking task. The spell-checking problem has received considerable attention in the past, and a variety of correction techniques have been proposed: in this context, noisy channel models (Church and Gale, 1991; Brill and Moore, 2000; Toutanova and Moore, 2002), to quote just a few, constitute one of the predominant and most successfull approaches. Under this paradigm, correction is performed on a word-per-word basis, and concerns primarily out-of-vocabulary tokens: the general assumptions are that most words are correctly spelled, and that in-vocabulary words should preferably be left un-

[2]A couple of on-line SMS-to-English translation systems are accessible on the Internet, see notably `http://www.transl8it.com/` and `http://www.lingo2word.com/`; “Netspeak” dictionaries, again for English, also abound. The situation is more or less comparable for French, see eg `http://www.traducteur-sms.com/`.

touched. In this context, the best correction(s) w of an erroneous word v is retrieved from the dictionary by combining the individual context independant probability of w with an error model probability, which computes the probability of mistyping v for w, based on the surface similarity between both forms. The key component here is the error model, which should not only capture orthographic similarities (Brill and Moore, 2000), but also phonetic similarities (Toutanova and Moore, 2002).

This framework easily extends to the case where several words should simultaneously be corrected: it is simply a matter of exploring the lattice of all possible corrections, which can be re-ranked using conventional tools such as statistical language models. This is basically the idea behind the reac system (Michel Simard, 2001),which recovers the correct accentuation of unaccented French texts using an error model, complemented with a statistical language model.

As far as SMS are concerned, this approach is essentially the one followed by (Choudhury et al., 2007), which models the joint probability of observing the word w represented by the character sequence $c_1...c_l$ by a Hidden Markov Model (HMM) whose topology takes into account both "graphemic" variants (typos, omissions of repeated letters, etc) and "phonemic" variants (i.e spellings that resemble the word's pronunciation). This HMM is initialized by considering the word's received orthography and phonology, with additional transitions to account for the possibility of inserting, substituting or deleting symbols. For the most frequent words in the corpus, the various parameters associated with these transitions are estimated on a training corpus; various heuristic are then used to plug these values in the HMMs that model the less frequent dictionary items.

2.2 The "translation" metaphor

A second approach to the problem consists in adopting the translation metaphor: using this analogy, the SMS language is just another foreign language and the normalization can be viewed as a "pure" machine translation (MT) task.

Using machine translation tools might be regarded as "an overkill" (Choudhury et al., 2007), considering the close relationships between source and target languages. Furthermore, learning the kinds of many-to-many correspondences between source and target sentences that make up for the high translation accuracy of phrase-based systems might be seen as introducing an unnecessary complexity, as SMS tend to be shorter, in terms of words, than their normalized counterparts. This suggests that looking for many (on the normalized side) to one (on the SMS side) might be good enough to capture most pairings. Finally, statistical machine translation tools incorporate mechanisms to model the possible mismatch in word order between source and target, which are virtually nonexisting when it comes to translating SMS.

This metaphor is, nonetheless, the one resorted to in (Aw et al., 2006), which uses a statistical phrase-based machine translation tool to convert English SMS texts into standardized English. This system incorporates some of the peculiarities of this translation task, which both simplifies the construction of the phrase-table and the decoding search algorithm. Using this system, (Aw et al., 2006) reports a 0.81 BLEU (Papineni et al., 2001) score on a set of 5,000 English SMS.

Normalization as translation is certainly a natural, and simple to implement, idea. Using phrase-based systems, it becomes possible to model (context-dependant) one-to-many relationships that are out-of-reach of the spell checking approach. We feel that it still overlooks some aspects of the task, notably the fact that the lexical creativity attested in SMS messages can hardly be captured in a static phrase table, where correspondences between SMS phrases and normalized phrases are learned by rote, rather than modeled.

2.3 The "speech recognition" metaphor

The SMS language has on occasions been described, sometimes abusively, as being closer to oral productions than to regular written texts. If we do not subscribe to this view, we nonetheless feel that a third metaphor is worth considering, that we call the "automatic speech recognition" (ASR) metaphor. This analogy stems from the fact that, for a significant fraction of tokens, the spelling of SMS forms tends to be a closer approximation of the phonemic representation of a word than of is its normative spelling.

In the speech recognition metaphor, an SMS message is thus primarily viewed as an alphabetic/syllabic approximation of a phonetic form. Given a suitable mechanism for converting the SMS stream into a phone lattice, the problem of SMS normalization becomes very similar to that of speech recognition, that is, the decoding of a word sequence in a (weighted) phone lattice. It actually becomes a much simpler problem, as (i) the acoustic ambiguity of speech input is typically much higher than the phonemic indeterminacy of SMS messages, (ii) some segmentation information is already available in the SMS text, which is in sharp contrast with (continuous) speech recognition, where word boundaries have to be uncovered.

Based on this general principle, we devised an ASR-like normalization system, which has three additional benefits: using a phonemic approximation provides the system with the ability to correct some (unintentional) typos; adopting an ASR-like architecture provides us with a "natural" framework for resegmenting agglutinated word forms; finally, in the larger context of SMS-to-speech applications, which is one of our targeted applications, the computation of a phonemic representation of the message can prove extremely valuable.

3 Two normalization systems

3.1 The MT-like system

Our first normalization system is entirely based on open-source, public domain packages for statistical machine translation. Giza++ (Och and Ney, 2003) is used to induce, based on statistical principles (Brown et al., 1990), an automatic word alignment of SMS tokens with their normalized counterparts; Moses (Koehn et al., 2007) is used to learn the various parameters of the phrase-based model, to optimize the weight combination and to perform the translation using a multi-stack search algorithm; the SRI language model toolkit (Stolcke, 2002) is finally used to estimate statistical language models. For this system, the training set has been split in a learning set[3] (approximately 25000 messages) and a development set (about 11700 messages), which is used to tune parameters.

As suggested in the previous sections, we have constrained both systems to consider only "monotonic" alignments between the source and the target languages.

3.2 The ASR-like system

In a nutshell, our second normalization system mimics the behavior of speech recognition system and decodes SMS message through a non-deterministic phonemic transduction; based on preliminary experiments, this simple architecture was augmented by an additional mechanism which specifically deals with out-of-vocabulary tokens.

In the following, we denote Λ the set of alphabetic symbols, Π the set of phonemic symbols, and Ω the set of lexical items. Using these notations, our architecture can be described as a pipe-line of the following components:

- the first processing step consists in a dictionary-based grapheme-to-phoneme conversion of some highly idiosyncratic forms, which deals with tokens[4] whose spelling in SMS does not reflect the phonemic content of the corresponding lexical item(s). This is, for instance, the case for common abbreviations (eg. *btw* for *by the way*) and for instances of "consonantic" spellings. The dictionnary used in the experiments reported above contains about 4,200 entries.

  This module is implemented as a finite-state transducer E which transduces letter sequences in Λ^* into mixed grapheme and phoneme sequences (in $(\Lambda \cup \Pi)^*$).

- the second module converts the graphemic portions of the input message into a phonemic string using a set of manually encoded *non-deterministic* letter-to-phone rules; these rules notably encode the possibility for each symbol to encode its spelling (eg. *u* for /ju/ or *R* for /ər/). Our system currently comprises about 150 letter-to-phone rules. The output of this module is a phone lattice, which represents all the possible pronunciations of the complete input stream.

  This module is also implemented as a finite-state transducer P representing a rational relation between $(\Lambda \cup \Pi)^*$ and Π^*: each grapheme-to-phoneme rule is compiled into a finite-state transducer; these individual rules are then, once properly ordered, combined through the composition operator. The resulting finite-state machine is denoted P.

- this phone lattice is then turned into a word based lattice, using an inverted pronunciation dictionary, which registers the known associations between phone sequences and words. This inverted dictionary contains approximately 21K words, which are the most

[3]See section 4 for a description of the corpora.

[4]At this stage, we take advantage of usual word separators to identify tokens in the message.

frequent words in our reference training corpus. This module is also implemented as a finite-state transducer D, which maps sequences in Π^* to sequences in Ω^*.

A key aspect of this module is its ability to alter the original tokenization, by freely inserting word separators whenever a phonetic word is recognized, no matter whether it corresponds to a complete input token or not. This mechanism is illustrated on Figure 1. As a result, this system can bear with agglutinations (i.e. absence of one or several word separators) in the input sequence.

- the final processing step consists in searching the word lattice for the most probable word sequence, as computed by a statistical language model (here, a smoothed n-gram language model estimated on the training corpus). Here again, the entire process is computed through finite state operations: the output language of the previous steps is intersected with the stochastic language model S (a weighted finite-state acceptor), and the most likely path is computed through dynamic programming.

As mentioned earlier, each module is implemented as a finite state acceptor or transducer; these modules are built and combined using tools from the FSM (Mohri et al., 1998) and the GRM (Allauzen et al., 2005) toolkit. As a result, the entire normalization process is computed by a weighted transducer $(E \circ P \circ D \circ S)$, which can be optimized off-line as is commonly done in finite-state speech recognition systems (Mohri and Riley, 1998).

In addition to these four main modules, the preprocessing module of the ASR-like system contains a number of small enhancements that improve the normalization of dates and hours. We furthermore had to modify the processing of out-of-vocabulary words: in the architecture sketched above, any word that does not belong to the vocabulary has to be decomposed into smaller, known, words, causing systematic errors. Our final ASR-like system allows these forms to be either decomposed phonetically or copied verbatim in the output. A complete description of this system is given in (Kobus et al., 2008).

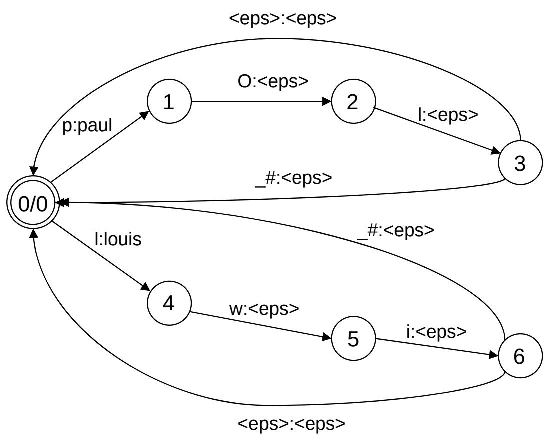

Figure 1: Transducing phone sequences into word sequences with a dictionary

This simplistic inverted dictionary recognizes two phonemic sequences: /lwi/ (for Louis) and /pOl/ (for Paul). Upon recognition of any such sequence, two transitions loop back to the initial state: one carries the input symbol '#', which is used whenever a word separator is encountered; the other is an ε transition, which allows to re-segment the input stream.

4 Experiments

4.1 Experimental protocol

The experiments reported below use two corpora. The first one has been collected at the University of Aix-en-Provence (Hocq, 2006); it contains approximately 9700 messages. The second corpus has been gathered in Belgium by the Catholic University of Louvain, and totals about 30000 messages (Fairon and Paumier, 2006). Both corpora contain, for each message, a reference normalization which has been produced and validated by human annotators. Both corpora were merged, lowercased, stripped from punctuation signs and standardized (in particular with respect to the anonymization conventions). This database was split in a training set (about 36700 messages) and a distinct test set of about 3000 messages. The training set was used both to train and tune the MT-like system and to estimate a 3-gram language model required in both approaches, using standard back-off procedures. Some relevant statistics regarding the sub-corpora that were used for training are given in Table 4.1.

For the evaluation, contrarily to (Aw et al., 2006; Guimier de Neef et al., 2007), who by analogy

	Aix	Louvain	Total
# messages	8,700	28,000	36,700
Original messages			
avg. length	14.3	21.6	19.9
# tokens	124 700	606 100	730 800
# types	13 600	37 900	43 600
% unknown	43.7 %	30.4 %	32.7 %
Normalized messages			
avg. length	15.4	23.2	21.3
# tokens	133 800	650 100	783 900
# types	8 200	20 800	23 300

Table 1: Statistics of the training corpora

Statistics on the original messages are computed after preprocessing (punctuation removal, etc.); the length of a message is the number of tokens; % unknown is the percentage of tokens that do not occur in the normalized message.

with the machine translation task, assess their system with the BLEU metric (Papineni et al., 2001), we decided to measure the performance of our normalization tool with the Word Error Rate (WER) and Sentence Error Rate (SER) metrics. This choice is motivated by the fact that the outcome of the normalization process is _ notwithstanding a couple of arbitrary normalization decisions _ almost deterministic, and does not warrant the use of BLEU, which is more appropriate to evaluate tasks with multiple references. Additionally, we feel that error rates are easier to interprete than BLEU values; for the sake of comparisons, some BLEU scores will nonetheless be reported.

4.2 Baseline results

In a first series of experiments, we evaluate our two systems and analyze their respective strengths and weakness. Table 4.2 reports the results of these experiments; the line 'initial' gives the corresponding numbers for the original messages, which gives a rough idea of the number of words that must be modified. As these results demonstrate, the MT-like system proves to be much more accurate than the ASR-like system.

Looking at the errors, the main problem with the latter system stems from the loss of the original spelling and tokenization information incurred during the grapheme-to-phoneme conversion step: as a consequence, many words that were correctly spelled in the input message are erroneously resegmented and decoded. This is evidenced by the high

	Sub	Ins	Del	WER	SER
initial	33.23	0.42	8.54	42.18	91.39
ASR-like	11.94	2.21	2.36	16.51	76.05
MT-like	7.34	0.71	4.22	12.26	63.41

Table 2: Evaluation of the baseline systems

Columns 'Sub', 'Ins', and 'Del' report respectively the number of substitution, insertion and deletion errors at the word level.

number of substitutions and insertions produced by this system, a large number of which concern function words. This phenomena is accentuated by the excessive liberality of grapheme-to-phoneme rules. For instance, to account for the erratic use of accentuated letters in SMS, the most general pronunciation rule for letter *e* (incidentally, *e* is the most frequent letter in French) predicts five pronunciations: /ə/, /e/, /ɛ/, /œ/, /ø/, plus the possibility of being deleted. As a result, first group verbal forms such as *aime* ('I or he/he love(s)') yield (at least) six different pronunciations, which result in many more combinations after resegmentation, such as *aime* ('(I, you, he/she) love(s)'), *aimé* ('loved'), *aimer* ('to love'), *aime et* ('I love and), *aime est* (I love is) etc. The same occurs with *de* ('of (the)', 'some-SING') and *le* ('the-SING'), which, through the phonemic encoding/decoding, become systematically ambiguous with their corresponding plural *des* and *les*. Sorting out the correct combination seems to be too hard a task for the statistical language model, since all these hypotheses include very high frequency tokens.

The MT-like system is significantly less error-prone: an error analysis reveals that the most common errors concern the insertion or deletion of function words, which can be attributed to noisy alignments in the phrase table. Another frequent source of error stems from agglutinated forms, notably combinations of clitic(s)+verb (e.g. *jtombrai* for *je tomberai* ('I will fall'), *jrentr* for *je rentre* ('I am coming back')) or (preposition and/or article)+noun or verbs (e.g. *cours 2droit* for *cours de droit* ('law class'), *dsortir* for *de sortir* ('to go out')...): whenever these forms are met in the training corpus, they can be correctly decoded; however, many novel forms only occur in the test set, owing to the fact that these types of agglutinations are "productive" (in the appropriate context). Contrarily to the findings of (Choudhury et al., 2007), which considered English messages, our corpus study reveals that this phenomena is far from marginal, and is a systematic source of errors for the MT system. It is noteworthy that about 17% of the tokens in the test SMS corpus do not occur in the training set, when the "true" out-of-vocabulary rate (computed on the reference messages) is only about 2.1 %.

Both systems are finally at pain to correctly recover the right number/gender/tense agreement, which is a general problem with n-gram language models in French, aggravated here by some irreducible indeterminacy: should '*désolé*' (masc.) in '*je suis désolé*' be corrected as '*désolée*' (fem) ? ultimately, this depends on the sex of the sender, which may be deduced from some other, potentially long distant, part of the message; the same indeterminacy occurs with the normalization of '*1*', which can be mapped to '*un*' (masc.) or '*une*' (fem); with '*aurai*' ('I will have'), which can be mapped with '*aurai*' or '*aurais*' ('I would have'), etc.

4.3 System combination

The analysis of normalization errors reveals that both systems have different strengths and weaknesses, suggesting that they could be used in combination. Indeed, oracle selection of the best output on a per message basis would yield an overall 9,63 WER, about 2.5 points absolute below the performance of the MT-like system.

Various ways to combine both approaches have been considered: we eventually decided to use the MT-like system for producing a first normalization; out-of-vocabulary tokens in the original SMS appear untouched in this output. For each of these, we use the ASR-like system to produce a series of "local" hypothesis, which are combined in a word lattice. This lattice is rescored with the statistical language model to yield the final output. This simple combination proved to yield significant improvements, decreasing the word-error rate from 12.26 to 10.82. The corresponding BLEU score is close to 0,8, in line with the findings of (Aw et al., 2006) for English, and comparing favorably with the 0.68 score reported in (Guimier de Neef et al., 2007) (for French, using a different test bed). Preliminary experiments suggest that using n-best list outputs from Moses instead of just the one best could buy us an small additional WER decrease.

The typical improvements brought by the combined system are illustrated by the following example, where two cases of agglutinated word forms are corrected, resulting in a correct output:

SMS	*oublié2tdir: tom a pomé c foto dlui en string*
MT-like	*oublié2tdir tom a paumé ses photos dlui en string*
Combined	*oublié de te dire tom a paumé ses photos de lui en string*
	(I) forgot to tell you (that) tom lost photos of himself in a thong

5 Conclusion and Perspectives

In this paper, we studied various ways to address the problem of normalization of SMS, by drawing analogy with related NLP problems, and accordingly reusing as much as possible existing tools or modules. Following (Aw et al., 2006), we found that using off-the-shell statistical MT systems allows to achieve very satisfactory WER; combining this system with a system based on an analogy with the speech recognition problem yields an additional 1.5 absolute improvement in WER. As it stands, our statistical normalization system seems to be sufficiently efficient to be used for text mining purposes; it also provides a useful tool to quantitatively analyze the various mechanisms involved in SMS spelling. The problem nonetheless remains far from being solved: our best system still makes at least one error on about 60% of the test messages.

There are a number of obvious improvements we might consider, such as using more accurate grapheme-to-phoneme rules, or plugging in a larger statistical language model, but we feel these would buy us only small increase in performance. As a first step to improve our normalization system, we would rather like to combine the existing approaches with a spell-checking approach. The most natural way to proceed would be to devise an alternative letter-to-word finite-state transducer C, aimed at converting space separated sequences of alphabetic symbols to the corresponding sequence of words, allowing for usual spelling errors (deletion/insertion of a letter, substitution, etc). Using the notations of section 3.2, the normalization system would thus be computed by the following finite state machine: $([E \circ P \circ D] \cup C) \circ S$.

Another natural extension would be to make this finite-state transducer stochastic: again, this would be a rather simple matter to train this transducer using the forward-backward algorithm (see (Jansche, 2003)) on the available training data.

6 acknowledgments

The authors wish to thank E. Guimier de Neef for providing us with one of the databases and other useful resources. Many thanks to our anonymous reviewers for helpful comments.

References

Allauzen, Cyril, Mehryar Mohri, and Brian Roark. 2005. The design principles and algorithms of a weighted grammar library. *International Journal of Foundations of Computer Science*, 16(3):403–421.

Anis, Jacques. 2001. *Parlez-vous texto ? Guide des nouveaux langages du réseau*. Éditions du Cherche Midi.

Aw, Aiti, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for SMS text normalization. In *Proceedings of COLING/ACL*, pages 33–40.

Brill, Eric and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Hong Kong.

Brown, Peter F., John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Journal of Natural Language Engineering*, 16(2):79–85.

Choudhury, Monojit, Rahul Saraf, Vijit Jain, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. In *Proceedings of the IJCAI Workshop on "Analytics for Noisy Unstructured Text Data"*, pages 63–70, Hyderabad, India.

Church, Kenneth W. and William Gale. 1991. Probability scoring for spelling correction. *Statistics and Computing*, 1:91–103.

Crystal, David. 2001. *Language and the Internet*. Cambridge University Press.

Fairon, Cédrick and Sébastien Paumier. 2006. A translated corpus of 30,000 French SMS. In *Proceedings of LREC 2006*, Genoa, Italy.

Fairon, Cédrick, Jean René Klein, and Sébastien Paumier. 2006. *Le langage SMS*. UCL Presses Universitaires de Louvain.

Guimier de Neef, Émilie, Arnaud Debeurme, and Jungyeul Park. 2007. TILT correcteur de SMS : évaluation et bilan quantitatif. In *Actes de TALN*, pages 123–132, Toulouse, France.

Hocq, S. 2006. étude des sms en français : constitution et exploitation d'un corpus aligné sms-langue standard. Technical report, Université Aix-Marseille.

Jansche, Martin. 2003. *Inference of string mappings for language technology*. Ph.D. thesis, Ohio State University.

Kobus, Catherine, François Yvon, and Géraldine Damnati. 2008. Transcrire les SMS comme on reconnaît la parole. In *Actes de la Conférence sur le Traitement Automatique des Langues (TALN'08)*, pages 128–138, Avignon, France.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic.

Michel Simard, Alexandre Deslauriers. 2001. Real-time automatic insertion of accents in French text. *Natural Language Engineering*, 7(2):143–165.

Mohri, Mehryar and Michael Riley. 1998. Network optimisation for large vocabulary speech recognition. *Speech Communication*, 25(3):1–12.

Mohri, Mehryar, Fernando Pereira, and Michael Riley. 1998. A rational design for a weighted finite-state transducer library. *Lecture Notes in Computer Science*, (1438).

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.

Stolcke, Andreas. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Langage Processing (ICSLP)*, volume 2, pages 901–904, Denver, CO.

Toutanova, Kristina and Robert Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 144–151, Philadelphia, PA.

Véronis, Jean and Émilie Guimier de Neef. 2006. Le traitement des nouvelles formes de communication écrite. In Sabah, Gérard, editor, *Compréhension automatique des langues et interaction*, pages 227–248. Paris: Hermès Science.