Statistical Machine Translation of Texts with Misspelled Words

Nicola Bertoldi **Mauro Cettolo** **Marcello Federico**
FBK - Fondazione Bruno Kessler
via Sommarive 18 - 38123 Povo, Trento, Italy
{bertoldi,cettolo,federico}@fbk.eu

Abstract

This paper investigates the impact of misspelled words in statistical machine translation and proposes an extension of the translation engine for handling misspellings. The enhanced system decodes a word-based confusion network representing spelling variations of the input text.

We present extensive experimental results on two translation tasks of increasing complexity which show how misspellings of different types do affect performance of a statistical machine translation decoder and to what extent our enhanced system is able to recover from such errors.

1 Introduction

With the widespread adoption of the Internet, of modern communication, multimedia and mobile device technologies, the amount of multilingual information distributed and available to anyone, anywhere, has exploded. So called social media have rapidly reshaped information exchange among Internet users, providing new means of communication (blogs, tweets, etc.), collaboration (e.g. wikis), and sharing of multimedia content, and entertainment. In particular, social media have today become also an important market for advertisement as well as a global forum for consumer opinions (Kushal et al., 2003).

The growing spread of user-generated content is scaling-up the potential demand for on-line machine translation (MT) but also setting new challenges to the field of natural language processing (NLP) in general. The language written and spoken in the social media presents an impressive variety of content and styles (Schler et al., 2006), and writing conventions that rapidly evolve over time. Moreover, much of the content is expressed in informal style, that more or less violates the standard grammar, contains many abbreviations and acronyms, and finally many misspelled words. From the point of view of MT, language of social media is hence very different from the one represented in the text corpora nowadays available to train statistical MT systems.

Facing all these challenges, we pragmatically scaled down our ambition and decided to investigate a basic, somehow preliminary, well defined problem: the impact of misspelled words in statistical MT. Unintentional typing errors are indeed remarkably frequent in online chats, blogs, wikis, reviews, and hence constitute a major source of noise (Subramaniam et al., 2009).

In this paper we aim at studying performance degradation of statistical MT under different levels and kinds of noise, and at analyzing to what extent statistical MT is able to recover from errors by enriching its input with spelling variations.

After a brief overview of NLP literature related to noisy texts, in Section 3 we consider different types of misspellings and derive simple but realistic models that are able to reproduce them. Such models are then used to generate errors in texts passed to a phrase-based statistical MT system. Next, in Section 4 we introduce an extension of a statistical MT system able to handle misspellings by exploiting confusion network decoding (Bertoldi et al., 2008).

Experiments are reported in Section 5 that in-

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 412–419,
Los Angeles, California, June 2010.

vestigate the trade-off between complexity of the extended MT decoder versus translation accuracy. Moreover, as the proposed model for handling misspellings embeds specific assumptions on how errors are generated, we also measure the robustness of the enhanced MT decoder with respect to different noise sources. Experiments are reported on two tasks of different complexity, the translation of Europarl texts and weather bulletins, involving English and Italian languages.

2 Previous Work

Most contributions addressing NLP of noisy user-generated content are from the text mining community. A survey about the different types of noise that might affect text mining is in (Subramaniam et al., 2009), while an analysis of how noise phenomena, commonly occurring in blogs, affect an opinion mining application is in (Dey and Haque, 2009).

Concerning spelling correction literature, many works apply the noisy channel model which consists of two components: a source model (prior of word probabilities) and a channel (error) model, that accounts for spelling transformations on letter sequences. Several approaches have been proposed under this framework, that mainly differ in the employed error model; see for example: (Church and Gale, 1991), (Brill and Moore, 2000) and (Toutanova and Moore, 2002).

Comprehensive surveys on methods to model and recover spelling errors can be found in (Kukich, 1992) and (Pedler, 2007); in particular, the latter work is specifically centered on methods for correcting so-called real-word errors (cf. Section 3). The detection of errors and the suggestion of corrections typically rely on the availability of text corpora or human-made lexical resources. Search for correct alternatives can be based on word similarity measures, such as the edit distance (Mitton, 1995), anagram hashing (Reynaert, 2006), and semantic distance based on WordNet (Hirst and Budanitsky, 2005). More sophisticated approaches have been proposed by (Fossati and Di Eugenio, 2008), that mixes surface and Part-Of-Speech Information, and (Schaback and Li, 2007), which combines similarity measures at the character, phonetic, word, syntax, and semantic levels into one global feature-based framework.

*a) *W *w had just come in from Australia [Australia]*
*b) good service we *staid one week. [Tahiti]*
*c) The room was *exellent but the hallway was *filty . [NJ]*
*d) is a good place to stay, if you are looking for a hotel *arround LAX airport. [Tahiti]*
*e) The staff was *freindly ... I was *conerned about the noise [CT]*

Table 1: Examples of misspellings found in on-line reviews of an hotel close to Los Angeles Int'l Airport. Corresponding corrections are: a) *We*, ϵ, b) *stayed*, c) *excellent*, *filthy*, d) *around*, e) *friendly*, *concerned*.

Concerning the literature of statistical MT, interest in noisy data has been so far considering issues different from misspelled words. For instance, (Davis et al., 1995) and (Vogel, 2003) address training methods coping with noisy parallel data, in the sense that translations do not perfectly match. Work on speech translation (Casacuberta et al., 2008) focused instead on efficient methods to couple speech recognition and MT in order to avoid error propagation. Very recently, (Carrera et al., 2009) conducted a qualitative study on the impact of noisy social media content on statistical and rule-based MT. Unfortunately, this work does not report any quantitative result, it is only based on a small selection of examples that are manually evaluated, and finally it does not address the problem of integrating error correction with MT.

3 Types of Misspellings

In general, a misspelled word is a sequence of letters that corresponds to no correctly spelled word of the same language (*non-word error*), or to a correct spelling of another word (*real-word error*). In the examples shown in Table 1, all marked errors are non-word errors, but the one in sentence b), which indeed is likely a misspelling of the word *stayed*.

Causes of a misspelling may be an unintentional typing error (e.g. **freindly* for *friendly*), or lack of knowledge about the proper spelling. Typing errors can originate from six different typing operations (Kukich, 1992): substitution, insertion, deletion, transposition, run-on, and split.[1] Lack of knowledge could be the cause of the misspelled **exellent* in sentence c).

[1] Run-on and split are the special cases of deleting and inserting blank spaces, respectively.

1. *your - you're*
2. *then - than*
3. *its - it's*
4. *to - too - two*
5. *were - where - we're*
6. *there - their - they're*
7. *a - an - and*
8. *off - of*
9. *here - hear*
10. *lose - loose*

Table 2: List of frequent real-word errors found in blogs. Source: http://www.theprobabilist.com.

An interesting combination of cause and effect is when lack of linguistic competence results in confusing the spelling of a word with the spelling of another word that sounds similarly (Hirst and Budanitsky, 2005). This could be likely the case of the Polynesian tourist that authored sentence b).

A short list of words frequently confused in blogs is reported in Table 2 while a longer list can be found in the Wikipedia.[2] Real-word errors typically fool spell checkers because their identification requires analyzing the context in which they occur.

In this paper, we automatically corrupt clean text with three types of noise described below. This procedure permits us to analyze the MT performance against different sources and levels of noise and to systematically evaluate our error-recovering strategy.

Non-word Noise We randomly replace words in the text according to a list of 4,100 frequently non-word errors provided in the Wikipedia. A qualitative analysis of these errors reveals that all of them originate by one or two keyboard typing errors of the kind described beforehand. Practically, non-word noise is introduced by defining a desired level of corruption of the source text.

Real-word Noise Similarly to the previous case, real-word errors are automatically introduced by another list of frequently misused words in the Wikipedia. This list contains about 300 pairs of confusable words to which we also added the 10 frequent real-word errors occurring in blogs reported in Table 2.

[2]See Wikipedia's "list of frequently misused English words".

Random Noise Finally, we may corrupt the original text by randomly replacing, inserting, and deleting characters in it up to a desired percentage.

4 Error-recovering Statistical MT

An enhancement of a statistical MT system is proposed with the goal of improving robustness to misspellings in the input. Rrror recovery is realized by performing a sequence of actions before the actual translation, which create reliable spelling alternatives of the input and store them into a compact word-based Confusion Network (CN).

Starting from the possibly noisy input text, spelling variations are generated by assuming that each character is a potential typing error, independent from other characters.

The variants are represented as a character-based CN that models possible substitutions, insertion, deletions of each character, with an empirically determined weight assigned to each alternative. The network is then searched by a non-monotonic search process that scores possible character sequences through a character n-gram language model, and outputs a set of multiple spelling variants that is finally converted into a word-based CN. The resulting word-based network is finally passed to the MT engine. In the following, more details are provided on the augmented MT system with the help of Figure 1, which shows how the system acts on the corrupted example "all off ame", supposed to be "hall of fame".

Step 1 The input text (**a**) is split into a sequence of characters (**b**) including punctuation marks and blank spaces (␣), which are here considered as standard characters. Moreover, single characters interleaved with the conventional empty character ϵ.

Step 2 A CN (**c**) is built by adding all alternative characters of the keyboard to each input character, including the space character ␣ and the empty character. When the string character is ␣, the only admitted alternative is ϵ. Possible alternative spellings of the original string correspond to paths in the CN. Notice that each CN column beginning with a standard character permits to manage insertion, substitution and split errors, while each column beginning with the empty character permits to handle deletion and run-on errors.

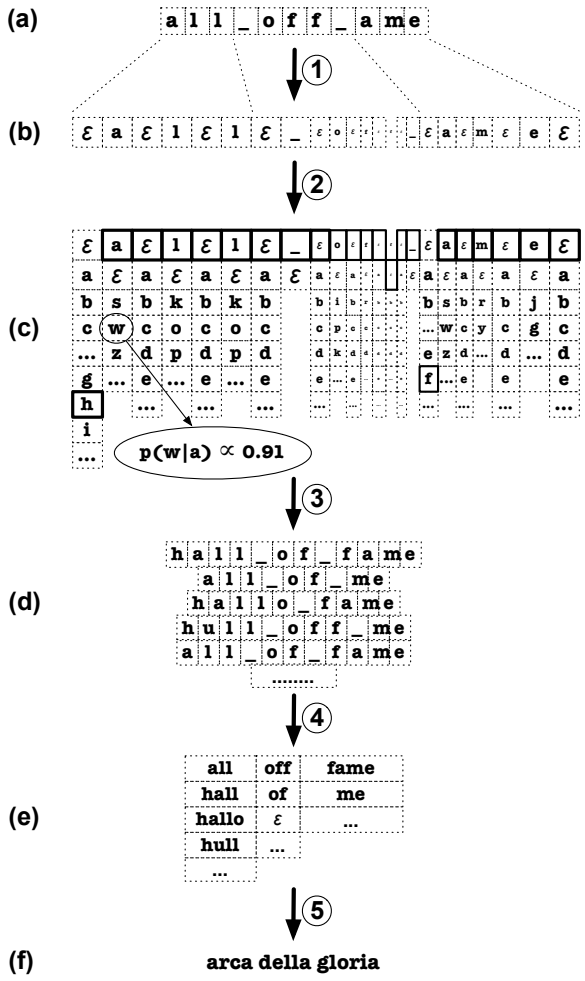

Figure 1: The whole process to translate the mistaken input "*all off ame* [*hall of fame*]" into "*arca della gloria*".

A probability distribution of confusable keystrokes is generated based on the distance between the keys on a standard QWERTY keyboard. This distribution is intended to model how a spelling error is actually produced. Hence, character alternatives in the CN are associated to a probability given by:

$$p\left(x|y\right) \propto \cdot \frac{1}{k \cdot d(x,y) + 1} \qquad (1)$$

where $d(x, y)$ is the physical distance between the key of x and the key of y on the keyboard layout; for example, the character *a* has a distance of 3 from the character c on the considered keyboard layout. The free parameter k tunes the discriminative power of the model between correct and wrong typing. In this paper, k was empirically set to 0.1. The ϵ and $\_$ characters are assigned a default distance of 9 and 999 from any other character, respectively.

For the sake of clarity, the probability $p(w|a)$ of just one entry is reported in Figure 1.

Step 3 The generation of spelling variations (**d**) is operated by means of the same decoder employed for translation (see below), but in a much simplified configuration which does not exploit any translation model. It is designed to search the input character-based CN for the n-best character sequences which better "correct" the mistaken input. In Figure 1 the best sequence is marked by bold boxes (**c**), and the empty character ϵ is removed for the sake of clarity (**d**). This process relies only on the character-based 6-gram language model trained on monolingual data in the source language. It is worth noticing that the generated spelling alternatives may in principle still contain non-words, just because they are selected by a character-based language model, which does not explicitly embed the notion of word.

Transposition errors are modeled both (i) indirectly through consecutive substitutions with appropriate characters and (ii) directly by permitting some re-orderings of adjacent characters. Moreover, preliminary experiments revealed that the explicit handling of deletion and run-on errors by interleaving input characters with the empty character ϵ (Step 1) is crucial to achieve good performance. Although the size of the character-based CN doubles, its decoding time increases only by a small factor.

Step 4 The n-best character sequences (**d**) are transformed into a word-based CN (**e**) (Mangu et al., 2000). First, each character-based sequence is transformed into a unifilar word-based lattice, whose edges correspond to words and timestamps to the character positions. Then, the unifilar lattices are put in parallel to create one lattice with all spelling variations of the input text (**a**). Finally, a word-based CN is generated by means of the *lattice-tool* available in the SRILM Toolkit (Stolcke, 2002).

Step 5 Translation of the CN (**e**) is performed with the Moses decoder (Koehn et al., 2007), that has been successfully applied mainly to text translation, but also to process multiple input hypotheses (Bertoldi et al., 2008), representing, for example, speech transcriptions, word segmentations, texts with possible punctuation marks, etc. In general,

set		#sent.	English		Italian	
			#wrd	dict.	#wrd	dict.
EP	train	1.2M	36M	106K	35M	146K
	test	2K	60K	6.5K	60K	8.3K
WF	train	42K	996K	2641	994K	2843
	test	328	8789	606	8704	697

Table 3: Statistics of train/test data of the Europarl (EP) and the Weather Forecast (WF) tasks.

Moses looks for the best translation exploring the search space defined by a set of feature functions (models), which are log-linearly interpolated with weights estimated during a tuning stage.

The rationale of storing the spelling alternatives into a word-based CN instead of n-best list is twofold: (i) the CN contains a significantly larger number of variations, and (ii) the translation system is much more efficient to translate CNs instead of n-best lists.

5 Experiments

Extensive experiments have been conducted on the Europarl shared task, from English to Italian, as specified by the Workshop on Statistical Machine Translation of the ACL 2008.[3] Additional experiments were conducted on a smaller task, namely the translation of weather forecast bulletins between the same language pair. Statistics on texts employed in experiments are reported in Table 3.

For both tasks, we created evaluation data by artificially corrupting input text with the noise sources described in Section 3. The module for generating spelling variations (Step 3) was trained on additional 4M and 16M running words in English and Italian, respectively.

We empirically investigated the following issues: (a) performance of the standard MT engine versus nature and level of the input noise; (b) performance of the error-recovering MT engine versus number of provided spelling variations; (c) portability of the approach to another task and translation direction; (d) computational requirements of the approach.

5.1 Impact of Noise

The first set of experiments involved the translation of corrupted versions of the Europarl test set. Figure 2 plots three curves of BLEU(%) scores, corresponding to different noise sources and noise ratios, given in terms of percentage of word error rate. It also shows the BLEU score on the original clean text. Notice that this baseline performance (25.16) represents the state-of-the-art[4] for this task.

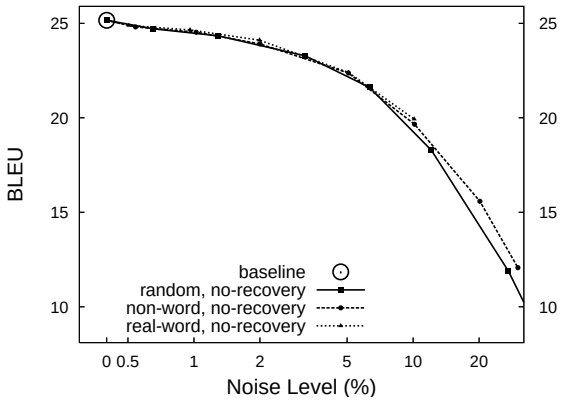

Figure 2: Translation performance as function of the noise level (in log-scale) for different types of noise.

The major outcome of these experiments is that the different types of errors seem to affect MT performance in a very similar manner. Quantitatively, performance degradation begins even for low noise levels – about 0.5 absolute BLEU loss at 1% of noise level – and reaches 50% when text corruption reaches the level of 30%. The similar impact of non-word and random errors is somehow expected. The plain reason is that both types of errors very likely[5] generate Out-Of-Vocabulary (OOV) words.

We find instead less predictable that the impact of real-word errors is indistinguishable from that of the other two noise sources. Notice also that most of the real-word errors produce indeed words known to the MT system. Hence, the question regards the behavior of the MT system when the sentence includes on OOV word or an out-of-context known word. Empirically it seems that in both cases the decoder produces translations with the same amount of errors. In some sense, the good news is that real-word errors do not induce more translation errors than OOV words do.

[3] http://www.statmt.org/wmt08/

[4] http://matrix.statmt.org/matrix

[5] Modulo noise in the parallel data and the chance that a random error generates a true word.

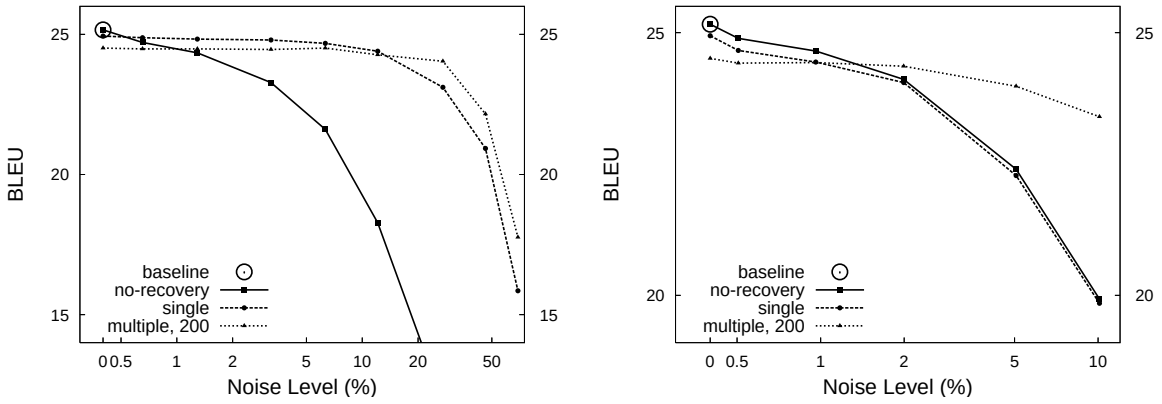

Figure 3: Performance of error-recovering method with random (left) and real-word (right) noise.

5.2 Impact of Multiple Corrections

Experiments presented here address evaluation of our enhanced MT system. In addition to nature and level of noise, translation performance is also analyzed with respect to the number (1 and 200) of spelling alternatives generated at Step 3. Figure 3 plots BLEU scores for random (left plot) and real-word (right plot) noises. For comparison purposes, the curves with no error recovery are also shown. Results with non-word noise are not provided since they are pretty similar to those with random noise.

It is worth noticing that real-word errors are recovered in a different way than random errors; in fact, for the latter a single spelling alternative seems sufficient to guarantee a substantial error recovery, whereas for real-word errors this is not the case.

Concerning the use of spelling variations, it is worth remarking that our system is able to fully recover from both random and non-word errors up to noise levels of 10%, which remains high even for noise levels up to 20%, where the BLEU degradation is limited to around 5% relative.

Real-word errors are optimally recovered in the case of multiple spelling variations until they do not exceed 2% of the words in the input text; after that, the decrement of the MT quality becomes significant but still limited to about 5% BLEU relative for a noise level of 10%. So the question arises about what could be a realistic real-word noise level. Clearly this question is not easy to address. However, to get a rough idea we can look at the examples reported in Table 1. These five sentences were extracted from a text of about 100 words (of which Table 1 only shows the sentences containing errors) that contain in total 8 errors: 7 of which are non-words and 1 is a real-word. Although from these figures reliable statistics cannot be estimated, a reasonable assumption could be that a global noise level of 10%[6] might contain a 1/10 ratio for real-word vs. non-word errors. Thus, looking at the real-word error curve of Figure 3, the inability to recover errors for noise levels greater than 2-5% should actually be acceptable given this empirical observation.

Another relevant remark from Figure 3 is that for low noise levels (less than 1%) the use of the error-recovering module is counterproductive, since it introduces more errors than those actually affecting the original input text, causing a slight degradation of the translation performance. If the computational cost to generate variants, which will be analyzed in the next paragraph, is also taken into account, it results evident the importance of designing a good strategy for enabling or disabling on demand the error-recovering stage. A starting point for defining an effective activation strategy is the estimation of the noise rate. For doing this, non-words can be counted by exploiting proper dictionaries or spell checkers; concerning real-word noise, its rate can be inferred either from the non-word rate, or by means of the perplexity, which is expected to become higher as the real-word error rate increases (Subramaniam et al., 2009). Once the noise level of the input text is known, the decision of activating the correction module can be easily taken on a

[6] By the way, at this noise rate, an error-recovering strategy would be highly recommended.

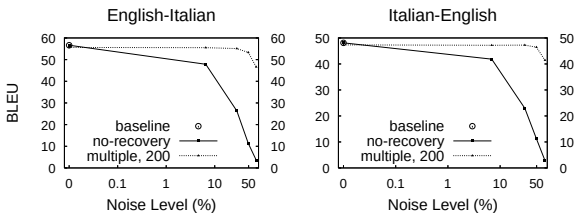

Figure 4: Effects of random noise and noise correction on translation performance for the WF task.

threshold basis. Alternatively, the proper working point, in terms of precision and recall, of the correction model could be dynamically chosen as a function of the actual noise level.

5.3 Computational Costs

Although our investigation does not address explicitly computational aspects of translating noisy input, nevertheless some general considerations can be drawn.

The effectiveness of our recovering approach relies on the compact representation of many spelling alternatives in a word-based CN. The CN decoding has been shown to be efficient, just minimally larger than the single string decoding (Bertoldi et al., 2008). On the contrary, in the current enhanced MT setting, the sequence of Steps 1 to 4 for building the CN from the noisy input text is quite costly. Rather than to an intrinsic complexity, this is due to our choice of creating a rich character-based CN in Step 3 for the sake of flexibility and to a naive implementation of Step 4.

5.4 Portability

So far we have analyzed in detail our approach on the medium-large sized Europarl task, for the English-to-Italian translation direction. For assessing portability, we also considered a simpler task –the translation of weather forecast bulletins– where the translation quality is definitely higher, for the same language pair but in both translation directions. The choice of the weather forecast task is not by chance. In fact, as the automatically translated bulletins are published on the Web, a very high translation quality is required, and then the presence of any typing error in the original text could be a concern. (By the way, for this task the presence of real-word errors is very marginal.)

Figure 4 plots curves of MT performance under random noise conditions against multiple spelling variations, for two translation directions. It can be noticed that the error-recovering system behaves qualitatively as for the Europarl task but even better from a quantitative viewpoint. Again, the recovering model introduces spurious errors which affect translation quality for low levels of noisy input, but in this case the break-even point is less than 0.1% noise level. On the other side, errors corrupting the input text are fully recovered up to 30-40% of noise levels, for which the BLEU score would be more than halved for non-corrected texts.

6 Future Work

There are a number of important issues that this work has still left open. First of all, we focused on a specific way of generating spelling variations, based on single characters, but other possible choices should be investigated and compared to our approach, like the use of n-grams of words.

An important open question regards efficiency of the proposed recovering strategy, since the problem has been only sketched in Section 5.3. It is our intention to analyze the intrinsic complexity of our model, possibly discover its bottlenecks and implement a more efficient solution.

Another topic, mentioned in Section 5.2, is the activation strategy of the misspelling recovery. Some further investigation is required on how its working point can be effectively selected; in fact, since the enhanced system necessarily introduces spurious errors, it would be desirable to increase its precision for low-corrupted input texts.

7 Conclusions

This paper addressed the issue of automatically translating written texts that are corrupted by misspelling errors. An enhancement of a state-of-the-art statistical MT system is proposed which efficiently performs the translation of multiple spelling variants of noisy input. These alternatives are generated by a character-based error recovery system under the assumption that misspellings are due to typing errors.

The enhanced MT system has been tested on texts corrupted with increasing noise levels of three different sources: random, non-word, and real-word errors.

Analysis of experimental results has led us to draw the following conclusions:

- The impact of misspelling errors on MT performance depends on the noise rate, but not on the noise source.
- The capability of the enhanced MT system to recover from errors differs according to the noise source: real-word noise is significantly harder to remove than random and non-word noise, which behave substantially the same.
- The exploitation of several spelling alternatives permits to almost fully recover from errors if the noise rate does not exceed 10% for non-word noise and 2% for real-word noise, which are likely above the corruption level observed in many social media.
- Finally, performance slightly decreases when input text is correct or just mistaken at a negligible level, because the error recovery module rewards recall rather than precision and hence tends to overgenerate correction alternatives, even if not needed.

Acknowledgments

This work was supported by the EuroMatrixPlus project (IST-231720), which is funded by the EC under the 7^{th} Framework Programme for Research and Technological Development.

References

N. Bertoldi, et al. 2008. Efficient speech translation through confusion network decoding. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1696–1705.

E. Brill and R. C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of ACL*. Hong Kong.

J. Carrera, et al. 2009. Machine translation for cross-language social media. http://www.promt.com/company/technology/pdf/machine_translation_for_cross_language_social_media.pdf.

F. Casacuberta, et al. 2008. Recent efforts in spoken language processing. *IEEE Signal Processing Magazine*, 25(3):80–88.

K. W. Church and W. A. Gale. 1991. Probability scoring for spelling correction. *Statistics and Computing*, 1(2):93–103.

M. W. Davis, et al. 1995. Text alignment in the real world: Improving alignments of noisy translations using common lexical features, string matching strategies and n-gram comparisons. In *Proceedings of EACL*, Dublin, Ireland.

L. Dey and S. M. Haque. 2009. Studying the effects of noisy text on text mining applications. In *Proceedings of AND*, pages 107–114, Barcelona, Spain.

D. Fossati and B. Di Eugenio. 2008. I saw tree trees in the park: How to correct real-word spelling mistakes. In *Proceedings of LREC*, Marrakech, Morocco.

G. Hirst and A. Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(01):87–111.

P. Koehn, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL - Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

K. Kukich. 1992. Spelling correction for the telecommunications network for the deaf. *Communications of the ACM*, 35(5):80–90.

D. Kushal, et al. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the WWW conference*, pages 519–528, Budapest, Hungary.

L. Mangu, et al. 2000. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer, Speech and Language*, 14(4):373–400.

R. Mitton. 1995. *English Spelling and the Computer (Studies in Language and Linguistics)*. Addison Wesley Publishing Company.

J. Pedler. 2007. *Computer correction of real-word spelling errors in dyslexic text*. Ph.D. thesis, University of London.

M. Reynaert. 2006. Corpus-induced corpus cleanup. In *Proceedings of LREC*, Genoa, Italy.

J. Schaback and F. Li. 2007. Multi-level feature extraction for spelling correction. In *IJCAI - Workshop on Analytics for Noisy Unstructured Text Data*, pages 79–86, Hyderabad, India.

J. Schler, et al. 2006. Effects of age and gender on blogging. In *Proceedings of AAAI-CAAW*, Palo Alto, CA.

A. Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP*, Denver, CO.

L. V. Subramaniam, et al. 2009. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of AND*, pages 115–122, Barcelona, Spain.

K. Toutanova and R. C. Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of ACL*, pages 144–151, Philadelphia, PA

S. Vogel. 2003. Using noisy biligual data for statistical machine translation. In *Proceedings of EACL*, Budapest, Hungary.