Correcting Input Noise in SMT as a Char-Based Translation Problem

Lluís Formiga **José A.R. Fonollosa**
TALP Internal Technical Report
Universitat Politècnica de Catalunya (UPC), Barcelona, 08034 Spain
October 31, 2012
{lluis.formiga,jose.fonollosa}@upc.edu

Abstract

Misspelled words have a direct impact on the final quality obtained by Statistical Machine Translation (SMT) systems as the input becomes noisy and unpredictable. This paper presents some improvement strategies for translating real-life noisy input. The proposed strategies are based on a preprocessing step consisting in a character-based translator (MT) from noisy into cleaned text. The use of a character-level translator allows us to provide various spelling alternatives in a lattice format to the final bilingual translator. Therefore, the final MT is the one that decides the best path to be translated. The different hypotheses are obtained under the assumption of a noisy channel model for this task. This paper shows the experiments done with real-life noisy input and a standard phrase-based SMT system from English into Spanish.

1 Introduction

Internet and Social Media have changed the trends of written text communication during the last years providing a straightforward and informal scenario (Agichtein et al., 2008). Thus, the focus of written text has evolved from grammatically correct structures to a content centered scenario. Nowadays, human web readers do not get surprised of finding misspellings or low-profile language. The text of chats, comments, tweets or SMS's is usually full of misspelled words, slang or wrong abbreviations introducing noise into the text data (Subramaniam et al., 2009; Yvon, 2010) and affecting NLP tasks such as text-mining, machine translation or opinion classification (Dey and Haque, 2009).

The Machine Translation (MT) task, as a field related to Natural Language Processing (NLP), is not immune to this noise (Aikawa et al., 2007). Generally, misspelling problems can be addressed with a simple Levenshtein distance under a noisy channel model paradigm (Brill and Moore, 2000). However, these algorithms are only based on a distance calculation process based on reference lexicons and have not been studied in detail in order to improve the final quality of MT. In contrast, Bertoldi et al. (2010) presented a preliminary work focused on preserving all spelling alternatives to the input of MT system through Confusion Networks (CNs) and hence involve the decoder the into the decision of which spelling alternatives are best, by means of its statistical learned models. However, this preliminary work was focused on an artificially generated noise that is no able to cover all the different properties of real-scenario weblog noise. In addition, their work modeled the alternatives heuristically without providing any training or adaptation step to the real data.

In this paper, we present a study of the performance of the aforementioned spelling correction strategies for real weblog translation requests. In addition, we present two new adaptive strategies based on obtaining the spelling alternatives from character-based translation models with multiple weighted cost functions.

In Section 2, we present the related work of SMT dealing with noisy data. The new misspelling recovery strategies are presented in Section 3. Experiments are carried out in Section 4 and discussed in Section 5. At last, general conclusions and future work are presented in Section 6.

2 Related work

Misspelling correction has been a recurrent issue to be resolved on NLP since its very first beginnings (Damerau, 1964). But it was not until the growth of computer resources that it became necessary to minimize noise on other computer-processed tasks such as Text-to-Speech synthesis (Kukich, 1992), Text Mining (Dave et al., 2003) or MT (Bertoldi et al., 2010). Good surveys of different types of noisy text and its related spell-correction programs can be found in (Pedler, 2007; Subramaniam et al., 2009) or the aforementioned (Kukich, 1992) along with (Mitton, 1996).

These surveys coincide on typifying the misspelling errors (character-based errors) in one of the following 4 classes: *a*) deletion, *b*) insertion, *c*) substitution or *d*) transposition errors. However, when moving from general misspelling towards chat/SMS contexts different types of errors emerge. These are high-level based errors (Subramaniam et al., 2009) such as *a*) general deletion of characters (e.g. message $\rightarrow$ msg), *b*) phonetic substitution (e.g. to $\rightarrow$ 2), *c*) abbreviation (e.g. lol $\rightarrow$ laughs out loud), *d*) slang, dialectical or informal usage (e.g, going to $\rightarrow$ gonna) and *d*) deletion of function words (e.g., I am driving back home $\rightarrow$ drvng hm). On this paper we focus into both low-level char-based errors continuing the work of Bertoldi et al. (2010) and higher level errors following the work of Contractor et al. (2010).

According to Deorowicz and Ciura (2005), misspelling correction methods can be separated as *isolated-word error detection-correction* methods, where isolated words are processed independently of their context and *context-dependent error detection-correction* methods where they feature their analysis in a more phrase-consistent manner.

Isolated methods perform a token-by-token independent analysis providing either one or multiple suggestions when they find a token likely to be a non-word. Some of the methods provide also a ranking from ordering the different suggestions by means of their probability to substitute the non-word token. This methods have evolved from simple edit distance techniques (Damerau, 1964) to probabilistic (Church and Gale, 1991) and phonetic similarity techniques (Philips, 2000) leaving several other techniques on its way such as key similarity or rule-based approaches. Recent advances have considered noisy channel approaches (Brill and Moore, 2000; Toutanova and Moore, 2002), neural networks or finite states automata (Deorowicz and Ciura, 2005). However these approaches fail on the generally known real-word problem. That is when an out-of-vocabulary word is replaced by a word in the dictionary that leads to grammatical inconsistencies throughout the sentence, e.g. I saw TREE trees in the park (Fossati and Di Eugenio, 2008).

In contrast, context-dependent methods seek the lexical/grammatical coherence of the sentence (Deorowicz and Ciura, 2005) but are more complex and language dependent as involve part-of-speech taggers, syntactic parsers or semantic role labeling (Hirst and Budanitsky, 2005). A good survey of these methods can be found on (Pedler, 2007) where they are divided as syntax-based approaches, confusion-sets based methods and semantic based approaches. Similar confusing-set approaches can be found using the Google search results instead of a Language Model (LM) (Jacquemont et al., 2007).

More recent methods use web-search queries (e.g. Google Web IT 3-grams) in order to overcome the problem of real-word spelling correction (Islam and Inkpen, 2009)

Among other new strategies, in this paper we study two already existing spelling correction strategies based on the Noisy Channel Model (Mays et al., 1991). First, we study the performance of a simple edit-distance based strategy computed from a lexicon of words under a noisy channel model scenario. Secondly, we study a strategy specially designed for the MT framework (Bertoldi et al., 2010). The second strategy is based on the generation of spelling alternatives through heuristically defined char-based Confusion Networks. We did not consider context-dependent strategies due to their dependency to several language-specific analysis tools, which are beyond the scope our study.

2.1 Edit-distance based spelling correction

The basic idea of this strategy is quite simple. First, the algorithm detects the words of the text that are missing in the lexicon of the translator (obtained from Model 1). For each one of the missing words, several alternatives are proposed consisting of the closest words (i.e. with a minor edit distance). In addition to the search of the original word, we address also the problem of word splitting (e.g. l iability) and word joining (e.g. thiscountry) searching alternatives for all the splits and joins possible of the original word. The different alternatives and the original sequence are combined in a lattice form that is submitted to the input translator (see Figure 1). The transition probabilities of the lattice are scored with the help of a 5-gram based language model of the source language.

2.2 Spelling correction based on the decoding of char-based confusion networks

Proposing the spelling alternatives to some closed lexicon has two main problems: *i*) the lack of ability to detect real-word misspellings (e.g three $\rightarrow$ tree) and *ii*) they are bounded to the lexicon available, which can ignore proper nouns or words borrowed from another language between others.

In contrast, some other papers (Brill and Moore, 2000; Toutanova and Moore, 2002) report significant improvement by modeling the probability of the character editing operations. These probabilities can be learned from some training data or otherwise they can be based upon a pronunciation model. These models assume the noisy channel model for the task.

Thereafter, a noisy-channel model based strategy adapted specifically to the task of MT was presented by Bertoldi et al. (2010). In addition they made a thorough study of the impact of the synthetic misspellings to the translation performance. Their method was inspired by the work on MT from speech (Bertoldi et al., 2008), where error propagation through different modules (ASR$\leftrightarrow$SMT) played a significant role on the performance of the overall system.

The strategy consists in generating hypotheses from a sequence of characters by means of confusion networks heuristically defined. The best sequences are retrieved from the CN according to char based language model (6-gram). The steps of the strategy are detailed next:

1. For each input sentence they generate a CN at char-level integrating the different spelling alternatives of the input text. The probabilities given to each alternative transition are based in a heuristically defined keyboard distance. The generation of the char sequences is interleaved by the empty-character notation symbol ε in order to solve misspellings caused by deletion and insertion mistakes. The keyboard distances are converted into probabilities as follows:

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

   where $d(x,y)$ is the physical distance between the key of x and the key of y on the keyboard layout, e.g. $d(a,c) = 3$. The free parameter k tunes the discriminative power of the model between correct and wrong typing.

2. The heuristically defined CNs are decoded through a simplified Moses decoder which relies solely on a simple character-based language model. The decoder provides N-best lists of character sequences. The decoding process also removes the empty-character symbols from the output.

3. The character-based N-best lists are transformed into a word-based CN (Mangu et al., 2000).

4. The word-based CNs are submitted as input to general Moses decoder (Koehn et al., 2007) that is able to process multiple input variants (Bertoldi et al., 2008). In the work they motivate the use of CNs instead of lattices since CNs they are more computationally efficient.

Using a simplified decoder to aid the translation task is not a novelty in machine translation tasks. A similar strategy is found in the recasing step when doing the post-processing, when a simplified single-word based phrase table is used in order to statistically learn which word has to be uppercased or otherwise remain lowercased on the final translation.

Regarding the simplified character-based decoder, only two probabilities come into play when generating the spelling variants: the ones from the layout distance model and the ones provided by the character-based LM. However the weights might be optimized. Thus, a tuning step is performed by an error minimization process, e.g. MERT (Och, 2003). To that purpose, at least a few hundreds of noisy sentences are needed along with their correspondent cleaned version by a human agent. In this paper we use a modified version of MERT that is capable to deal with Character Error Rate (CER) score to perform the optimization.

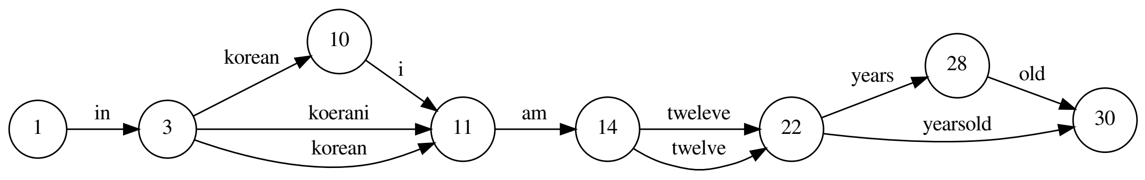


Figure 1: Lattice obtained when finding edit-distance based alternatives for the sequence "*in koerani am tweleve yearsold*".

3 Adaptive spelling correction based on character-based translation models

In Bertoldi et al. (2010) the spelling hypotheses are obtained only from the heuristically defined character-based CNs. The simplified decoder is based only on a single character-based LM without any phrase-based or distortion models. Hence, the strategy assumes that all editing operations are equally weighted at decoding stage since CNs are globally weighted (weight-i). However, state-of-the art decoders (e.g. Moses) may deal with multiple transformation models. We propose two new strategies that deal with multiple transformation models. The first strategy works with a heuristic phrase-table containing different model scores depending on the type of transformation that is addressed (i.e. identity, substitution, deletion, addition), and also allows the reordering of chars according to a distance-based distortion model. The second strategy is based on the classical SMT training strategy but adapted to character level.

These strategies allow weighting all the probability models independently. Thus, they are more suited for being adapted into training data by means of an optimization step as more functions take part into the final hypothesis.

Analogously to the previous approach, the N-best hypotheses may be fused in a lattice or confusion network form and submitted as input to the final translator. In this paper we only work with lattices as input to the translator. The lattices are built from a three-step process as follows: first each character-sequence of the N-best list is transformed into a single-path word-based lattice, then the different word lattices are aligned to the original sequence through a distance based algorithm. Once aligned, the single-path lattices are combined generating a single lattice containing all the spelling variations that have been seen on the N-best output of the character-based decoder.

As for assigning probabilities to the word-alternatives (edges) when building the final lattice, we considered the proportional presence of a word in the N-best list. That is if "twelve" is found 4 times in the same position in a 10-best list then its arriving edge gets a probability of p =0.4.

3.1 Misspelling correction through a heuristic phrase-table

All the possible edit operations can be represented through phrase table transformations. Therefore, our first strategy designs a heuristic phrase table with all the probabilities of the possible transformations separated in different models according to their type. A fragment of the table is given in Table 1. The table is composed of 4 transformation models: Identity, Substitution, Deletion and Addition.

Probabilities are given on an exponential base as Moses works on the log-space and we are more interested in working in a linear space. We assign a binary probability (e^0,e^1) to identity, addition and deletion operations because they are not distance based. On the other side, since substitution operations might be based in a distance model, we assign the same probability defined on Bertoldi et al. (2010) by equation 1. It is important to highlight that each entry of the phrase table takes a single non-zero probability for its related operation, being all the others set to e^0.

In addition, we also consider that transposition operations can be performed by the distance

Source	Target	Probabilities			
		Ident.	Subst.	Del.	Add.
a	a	e^1	e^0	e^0	e^0
a	b	e^0	$e^{p(b\|a)}$	e^0	e^0
a	_	e^0	$e^{p(\_\|a)}$	e^0	e^0
a	NULL	e^0	e^0	e^1	e^0
a	_ a	e^0	e^0	e^0	e^1
a	a _	e^0	e^0	e^0	e^1
a	a b	e^0	e^0	e^0	e^1
a	b a	e^0	e^0	e^0	e^1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 1: Heuristic phrase table used for the spelling hypotheses generator (Moses decoder) which separates probabilities of translations in different transformation (or translation) models considering either identity (Ident.), substitution (Subst.), deletion (Del.) or addition (Add.) operations. Note that the "_" symbol refers to an empty space and "NULL" refers to an empty character (deletion).

based reordering implemented in the Moses decoder. That approach contrasts with the CN decoding approach (Bertoldi et al., 2010), were transposition operations were performed by the sum of deletion and addition operations. In order to prevent big reorderings we limit the distortion up to three positions.

In summary, we consider 6 different probability models: character-based language model, distance based distortion, identity, substitution, deletion and addition.

3.2 Misspelling correction through character-based SMT models

With the strategies presented so far we have only addressed issues related to low-level misspellings. Unfortunately, as it has been mentioned, the noise of chat/SMS domains concerns higher level errors. Within these errors we can distinguish two types: em i) structural errors in the order of words within the sentence due to the lack of knowledge of the language and *ii*) on-purpose induced errors based on the economy of language consisting of abbreviations, acronyms, contraction or slang among others.

To address this latter case there is recent work based on the use of an SMT to convert noisy text into cleaned text (Contractor et al., 2010). This strategy allows the SMT to learn the most common conventions used in informal language. However, we did not found any research work related in learning these changes at character level in order to propose hypotheses.

Our second improvement strategy learns a SMT at character-level in order to propose alternative spelling to the final translator. In this sense, we first clean manually a certain amount of noisy text (e.g 8000 sentences) gathered from web translation requests. Afterwards, both the noisy text and the clean text are converted to character sequences using a common alignment tool (e.g. GIZA++). Once aligned, the character level bicorpus is used to learn the typical probabilities of a phrase-based SMT. That is: *i*) $\varphi(f|e)$ inverse phrase translation, *ii*) $lex(f|e)$ inverse lexical weighting, *iii*) $\varphi(e|f)$ direct phrase translation and *iv*) $lex(e|f)$ direct lexical weighting along with a *v*) transformation penalty (which is e^1) inspired in the phrase penalty.

The main difference of this strategy with respect to the one presented in Section 3.1 is the building of the phrase-table. While the previous strategy builds a heuristic phrase-table, the new one learns from the real proofreading. This approach also allows the use of a penalty model (based on word-based penalty of Moses).

In that case, we consider 8 different probability models: character-based language model, distance based distortion, $\varphi(f|e)$, $lex(f|e)$, $\varphi(e|f)$, $lex(e|f)$, transformation-based penalty and character-based penalty.

4 Experiments

We based our experiments under the framework of a factored decoder (Moses – Koehn and Hoang (2007)) from English into Spanish. Concretely, we translate the source words into target words plus their POS tags (Factored Moses from 0 to 0,2) using two separate language models for improving the fluency of the output. We aligned the corpus with stems through mGIZA (Gao and Vogel, 2008). The decoder was trained with the material from the WMT12 (Callison-Burch et al., 2012) MT Shared Task (See details in Table 2). We used the Freeling analyzer (Padró et al., 2010) to tokenize and POS-tag both sides of the corpus (English and Spanish). We preprocessed the text to lowercase in order to overcome the casing problems, which are quite fre-

Corpus		Sent.	Words	Vocab.	avg.len.
EPPS	Eng	1.90 M	49.40 M	124.03 k	26.05
	Spa		52.66 M	154.67 k	27.28
News.Com	Eng	0.15 M	3.73 M	62.70 k	24.20
	Spa		4.33 M	73.97 k	28.09
UN	Eng	0.83 M	20.57 M	183.40 k	24.54
	Spa		23.95 M	191.49 k	28.57

Table 2: Details of different corpora used for training the models. The counts are computed before lowercasing.

quent under noisy scenarios. In addition, we segmented the Spanish clitics (e.g. cómpramelo) and contractions (e.g. del) as separate words (compra-me-lo, de-el). We trained the language models (LM) with the SRILM Toolkit (Stolcke, 2002) at 5-gram level for words, 7-gram level for POS-tags and also 7-gram level for the character-based language models. The language models were built independently for each of the WMT12 monolingual corpora (Europarl, News, UN and Gigafrench) and then linearly combined towards reducing the perplexity of the noisy text. In the same way, the weights of the system were optimized by MERT and a BLEU score with the help of a weblog development set consisting of 999 sentences, as explained in the next section.

We have conducted the experiments in three parts. Firstly we studied the properties of the real-life noisy scenario. Then, we compared the systems performance when generating spelling correction hypotheses and then we analyzed the actual performance of the systems as for the translation task.

4.1 Real-life scenario: dealing with actual noisy words

Most of the work mentioned in Section 2, deals with synthetic or controlled noisy scenarios. However, real-life texts are poorly related with this controlled scenario in terms of literary quality. Text quality usually depends on the specific domain or environment which in turn implies different levels of orthographic and semantic correctness, or simply different use of vocabulary words or semantic expressions (Agichtein et al., 2008; Subramaniam et al., 2009).

As we wanted to deal with real data, we used weblog translations from the FAUST project (Pighin et al., 2012) for testing the translation performance with noisy texts. Regarding the weblog translations we considered 1997 translation requests submitted to Softissimo's portal [1]. Unlike already available resources, the data in this corpus reflects the real needs and requirements of casual users of translation systems, and covers a wide spectrum of domains and styles. Some requests are complete, well- formed sentences, whereas others are just snippets of text copied and pasted from somewhere else (e.g., chat rooms, web pages, software manuals, just to name a few), or simply words or noun phrases in isolation. In many cases, the input is disfluent or ungrammatical. In some cases, the interpretability of the input sentence is questionable. Nevertheless, real-life translation systems must be able to cope with this kind of data, and to produce outputs which, at the very least, should contain useful clues to satisfy practical needs of users. A full description is given in Pighin et al. (2012).

Two independent human translators corrected the most obvious typos and provided reference translations into Spanish for all of them along with the clean versions of the input requests. Thus, we consider three different test sets from this material:

1. *Weblog Raw (wr)* The noisy weblog input. It contains misspellings, slang and other input noise typical from chats, forums, etc. These translations are evaluated with their correspondent reference provided by each translator (two references).

2. *Weblog Clean$_i$ (w0 and w1)* The cleaned version of the input text provided by each translator on the source side. Cleaned versions may differ due to the interpretation of the translators. In general terms, w_0 fixes both high-level and low-level errors whereas w_1 is more bounded only to fix low-level errors. (e.g. If

[1] http://www.reverso.net

	Perplexity	
Data	DEV	TEST
Original Source (wr)	835.713	891.55
Clean Source 0 (w0)	541.58	533.74
Clean Source 1 (w1)	575.35	660.34
Combined Clean Sources (w0.w1)	558.39	594.03

Table 3: Perplexity obtained between original and cleaned data.

		WER	
Target	Reference	DEV	TEST
wr	w0	13.54%	16.33%
	w1	8.61%	6.51%
	w0,w1	6.67%	6.35%

Table 4: Word-error rate obtained between original and cleaned data. Note that $w0, w1$ stands for WER evaluated with two references (mWER).

you dont like to chat → If you don't want to chat — If you don't like chatting).

3. *Weblog Clean0.1 (w0.w1)* In that case we mix up the criteria of the different translators. In that case the cleaned versions are concatenated (making up a set of 3994 sentences) and evaluated with their respective translations (two references).

In order to perform the different optimization tasks, we have divided the noisy set in development (999 sentences) and test (998 sentences) sets.

In this Section we give some indicators of the presence of noise within the weblog data sets following the work performed by Subramaniam et al. (2009). Concretely we will measure the level of noise on the real data computing *Word-Error-Rate* (WER) (Kobus et al., 2008) and Language Model Perplexity (Kothari et al., 2009).

Perplexity results are detailed on table 3 whereas word-error-rate results are detailed on table 4. From the tables it can be observed that WER can vary up to 5% depending on the human translator who made the cleaning task. Still, considering both human translators, the averaged WER is around 11%, and no notable differences are found between the development and the test sets. In that sense, the *w0* set takes higher edit modifications than *w1* compared to the original text. Consequently, as for the perplexity results, *w0* takes less perplexity regarding the character-based LM with respect to *w1*. This fact shows that strong changes (due to high-lever error fixing) on the edit distance (higher WER) lead to a more normalized input (lower perplexity).

4.2 Implemented Systems

In our study we compare the different strategies presented in Sections 2 and 3. They are summarized next:

- *Distance*: This strategy searches for the spelling alternatives through the closest words (regarding a Levenshtein distance) within a lexicon. We used as a lexicon the IBM Model 1 source words from the factored decoder. Once the alternatives are combined through a lattice, the lattice is scored by means of a word-based language model. Then this lattice is passed to the decoder.

- *Confusion*: This strategy follows most part of the proposal of Bertoldi et al. (2010). The alternatives are obtained by decoding a heuristically defined confusion network at character level. We built a 7-gram based language model in order to perform the decoding of the CN. In our study we provide the alternatives through a lattice instead of a CN as the only motivation for using CNs was the reduction of computation time. It has two weights to be tuned (language model and CN edges).

- *Heuristic PT*: This strategy uses a heuristically defined phrase-table (also at character level) for finding the different spelling alternatives. In addition this strategy makes use of the distance based distortion model of Moses. The main advantage of this strategy is that the different types of transformation (Reordering, Identity, Substitution...) may be weighted independently according to some development text sets. Distortion was limited to maximum distance of 3 positions. It has 6 weights to be tuned (4 edit operations, language model and distortion).

- *GIZA PT*: This strategy learns the character transformation phrase-table from some training bicorpus previously aligned. In that case we post-edited manually 8000 noisy sentences submitted to the same portal (Softissimo), so they are similar to the dev/test sets. The number was chosen heuristically based on the previous work of Aw et al. (2006).The noisy and cleaned sentences were character-aligned with mGIZA and then the standard phrase-based SMT models were trained at character level. Distortion limit was set to the Moses standard 6-positions. The main advantage of this strategy is its capacity to deal with higher level errors such as acronyms, contractions or slang forms typical from the chat/SMS domain. It has 8 weights to be tuned (5 phrase-table model weights, language model, character penalty and distortion).

The weights of the character-based strategies were tuned with the weblog development set already mentioned. We modified the MERT script to work with the Character Error Rate metric.

Regarding the N-best size for building the lattice, we studied different values on the low-range in order to obtain low-dimensionality lattices. Thus we studied building the lattice from the 1-best, 5-best and 10-best lists of the preprocessing step.

Additionally, the fact of providing a lattice to the Eng$\rightarrow$Spa translator requires to perform a retuning step in order to find the appropriate weight value for the edges of the lattice (w_I). We did this retuning step for each strategy only searching different values for the w_I weight and fixing all the others to the already tuned value.

4.3 Spelling Correction Strategies Performance

Before evaluating the performance in the translation task, we wanted to evaluate the suitability of each strategy for finding good spelling alternatives. We did this evaluation either in the development and test weblog sets using four different evaluation metrics: CER, WER, BLEU and METEOR (Denkowski and Lavie, 2011). We left out of our study Precision/Recall analyses as we are focused on the translation performance and not only the misspellings, they could be considered in future work. These results were obtained by comparing the automatically cleaned input with the two human post-edited references (being CER and WER evaluated through mCER and mWER). In case of CER, WER and BLEU this comparison was done considering only the 1-best spelling alternative of the strategy. In case of METEOR we computed the oracle results considering the best hypothesis from the obtained N-best list (1000-best for dev and 50-best for test).

Results are detailed in table 5. Within these results "Baseline" refers to the case when no spelling correction strategy is applied at all. We observe that the GIZA PT strategy performs better when considering the 1-best output whereas the Heuristic PT strategy finds better alternatives within the N-best list, despite they are not the first hypothesis. In addition we can see that the Distance strategy worsens the baseline results for the 1-best tests whereas it can achieve a slightly improvement in the N-best based tests. These results seem to indicate that the language-model used for ranking the final hypothesis might not be fully functional for that purpose. We have to remember that the language model was built from the formal WMT12 data and thus the interpolation towards perplexity reduction may not be enough to obtain a good language model based on the open-domain of weblog requests.

4.4 Translation Task Performance

After evaluating the spelling correction strategies we evaluated the overall strategy involving the misspelling correction and translation tasks. For each studied system, we performed the translation with four different inputs: *i*) the original noisy source (*wr* – two references), *ii*) the high-level cleaned version (*w0* – one reference), *iii*) the low-level cleaned version (*w1* – one reference) and *iv*) their combination (*w0.w1* – two references).

We analyzed the results with BLEU and METEOR (Denkowski and Lavie, 2011). The results are detailed in Tables 6 and 7. In addition, the effects to the final translation of increasing the size of the N-best are plotted in Figures 2 and 3.

In general terms we observe that the GIZA PT strategy outperforms all the other strategies across all the metrics and test sets. Regarding the recovery from the noisy set (*wr*) we can see a maximum gain of 0.36 BLEU points and 0.4 METEOR points. Also we can observe slightly improvements on the clean sets: ~ 0.16 BLEU points and ~ 0.17 METEOR points. The improvements on the clean sets are explained by some tokenization errors of Freeling that are fixed thanks to the misspelling correction step (e.g. I'll go $\rightarrow$ I will go or I 'll go). In that sense the

Strategy	*dev*	*test*	*dev*	*test*	*dev*	*test*	*dev*	*test*
	CER		**WER**		**BLEU**		**METEOR nbest oracle**	
Baseline	3.41	3.09	6.67	6.35	90.62	90.24	63.10	63.17
Distance	3.47	3.19	6.92	6.96	89.87	89.02	64.63	63.62
Confusion	3.40	3.10	6.62	6.36	90.72	90.19	64.00	63.69
Heuristic PT	3.36	3.07	6.35	6.23	91.25	90.37	**65.81**	**64.92**
GIZA PT	**3.33**	**2.99**	**6.26**	**5.82**	**91.32**	**91.02**	64.02	64.24

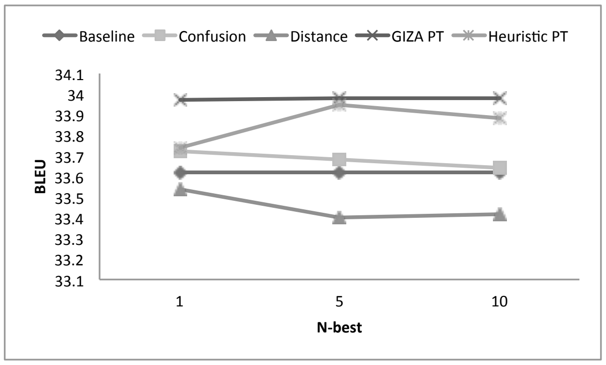
Table 5: CER/WER/BLEU/METEOR scores obtained when cleaning the texts. CER/WER/BLEU are computed over the first spelling alternative found on the N-best list generated by each system while the METEOR score is computed through the best spelling alternative found on the N-best list. The best results are depicted in bold.

Strategy	**N-best**	**w0**	**w0.w1**	**w1**	**wr**	**AVG**
Baseline	1	30.61	37.44	29.86	33.62	32.88
Distance	1	30.20	36.99	29.41	33.54	32.54
Distance	5	29.84	36.67	29.21	33.40	32.28
Distance	10	29.83	36.65	29.20	33.42	32.28
Confusion	1	**30.77**	37.56	29.90	33.72	32.99
Confusion	5	30.65	37.44	29.74	33.68	32.88
Confusion	10	30.59	37.35	29.66	33.64	32.81
Heuristic PT	1	30.70	37.51	29.83	33.74	32.95
Heuristic PT	5	30.45	37.27	29.62	33.95	32.82
Heuristic PT	10	30.37	37.17	29.50	33.88	32.73
GIZA PT	1	**30.77**	37.61	29.97	33.97	33.08
GIZA PT	5	30.76	37.62	29.98	**33.98**	33.09
GIZA PT	10	30.76	**37.63**	**30.00**	**33.98**	**33.09**

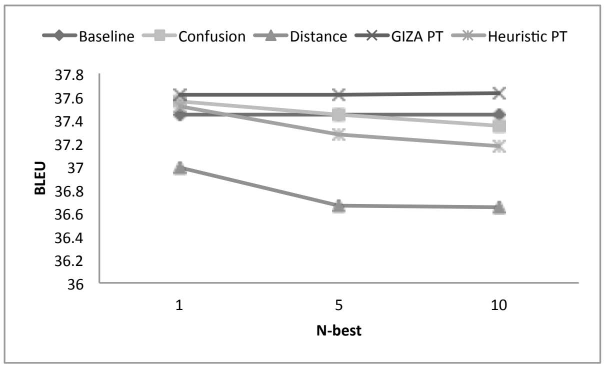
Table 6: BLEU scores obtained applying different misspelling MT strategies

Strategy	**N-best**	**w0**	**w0.w1**	**w1**	**wr**	**AVG**
Baseline	1	54.41	58.08	54.64	54.93	55.51
Distance	1	54.07	57.65	54.18	54.75	55.16
Distance	5	53.68	57.28	53.80	54.43	54.80
Distance	10	53.68	57.26	53.78	54.46	54.80
Confusion	1	54.56	58.20	54.72	55.04	55.63
Confusion	5	54.32	58.00	54.52	54.92	55.44
Confusion	10	54.26	57.92	54.45	54.87	55.37
Heuristic PT	1	54.51	58.14	54.64	55.08	55.59
Heuristic PT	5	54.15	57.76	54.30	55.05	55.31
Heuristic PT	10	53.98	57.56	54.12	54.89	55.14
GIZA PT	1	54.60	58.22	54.75	55.31	55.72
GIZA PT	5	**54.62**	58.24	54.77	55.33	55.74
GIZA PT	10	54.61	**58.24**	**54.79**	**55.33**	**55.74**

Table 7: METEOR scores obtained applying different misspelling MT strategies

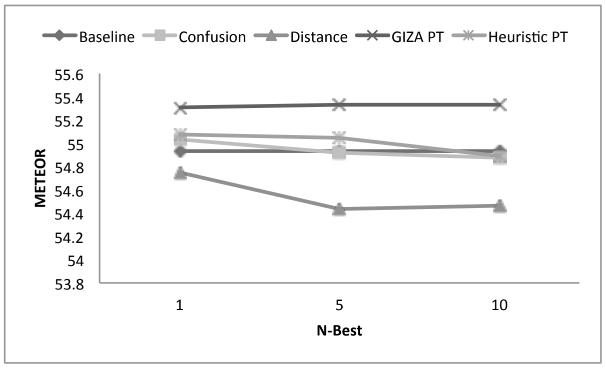


(a) Original Source (*wr*)

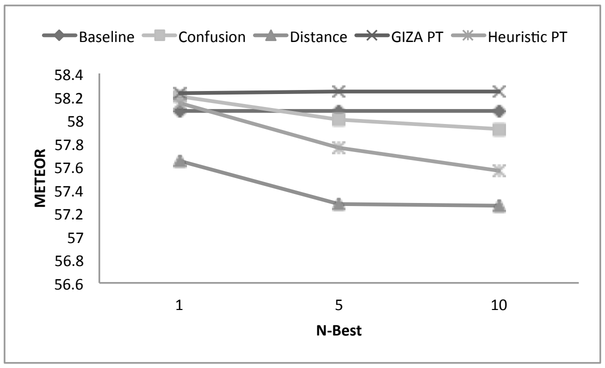


(b) Combined Clean Sources (*w0.w1*)

Figure 2: Performance in terms of BLEU of the different spelling translation strategies evaluated once translated.



(a) Original Source (*wr*)



(b) Combined Clean Sources (*w0.w1*)

Figure 3: Performance in terms of METEOR of the different spelling translation strategies evaluated once translated.

misspelling correction step also performs a revision of the tokenization carried out beforehand.

Observing the Figures we can see also that the GIZA PT strategy is quite robust while increasing the N-best list to build the lattice. In contrast, the other strategies decrease the quality when the N-best list size is increased. As it has been explained, this might be motivated due to the high perplexities of the language model to the open domain text, making it not suitable for ranking the different hypotheses.

The Confusion and Heuristic PT strategies perform slightly better than the baseline (no-processing at all) for the 1 and 5-best configurations in the noisy test sets. However, when it comes to the clean test sets they are not able to improve the baseline and worsening the result in case of increasing the n-best list size.

The Distance based strategy is the worst, even compared to the baseline, across all the metrics and test sets. Making it not feasible for dealing with noisy input translations.

5 Discussion

The results of the experiments allow us to gain an in-depth specific understanding of how each strategy contributes to the misspelling correction when making MT from real-life texts.

The translation results obtained are coherent with the 1-best spelling correction results reported in table 5. However, the higher scores obtained in the METEOR N-best oracle case show that there may be scope for improvement if a more adecuate language model based on an open domain (e.g. Google N-grams) helps in the reranking of the proposed hypotheses.

In detail, we see that strategies based on a simple distance with respect to some closed lexicon worsen the baseline system. This is explained by the real-word errors corrections and the lack of a good language model (perplexities are over 500). Replacing a misspelled word with a correctly spelled word but senseless in that specific context usually leads to a worse automatic translation.

Secondly, the results of the heuristic strategies (Confusion and Heuristic PT) show that the translation scores improve with noisy input but can decrease the quality of clean input translations. This behavior had already been identified by Bertoldi et al. (2008) in two cases: when the noise level was lower than 2% or when the errors were caused mainly by real-word errors. In order to avoid the decrease of the MT quality on clean texts for the heuristic strategies, they (Bertoldi et al., 2010) reported that it would be necessary to incorporate a noisy-text detector step on the input data which would trigger the correction process.

However, the new GIZA PT strategy presented in this paper is also robust to clean text, avoiding the need of a clean / noisy-text detector. In fact, the GIZA PT strategy can partially correct both the noisy and cleaned text fixing low-level (e.g. thats fun $\rightarrow$ that is fun) as well as high-level errors (e.g. prove'em wrong $\rightarrow$ prove them wrong).

In addition, we want to highlight that the presented methodology is somewhat language independent since it does not need deep-language tools such as parsers or semantic role labelers. A small training corpus (or development corpus in case of the heuristic strategies) of about 8000 sentences might be enough to obtain a good spelling corrector, given a constant noise density ratio bounded to weblog translations.

6 Conclusions and Future work

We presented a detailed study of different spelling correction strategies for improving the quality of Machine Translation in real-life noisy scenarios. Real-life errors may be produced by different causes such as general misspelling (low-level errors) or informal text conventions (high-level errors) among others.

Apart from the basic strategy based on the Levenshtein distance, we also studied two strategies based on heuristic models and a strategy based on building a character-level translator. Regarding the heuristic methods, we adapted an existing strategy to take full advantage of standard feature functions such as distortion and we included a MERT-based tuning of the weights.

Whereas the distance-based strategy is not able to deal with real-life errors, the heuristic strategies show some improvement to the baseline translation and are easy to implement. However, the heuristic strategies are bounded to low-level misspelling errors and rely solely in the quality of the language model used for scoring the different alternatives.

In contrast, the trainable character-based strategy, namely GIZA PT, reports a significant and robust improvement across all the evaluated test sets and metrics. The GIZA PT offers a good trade-off between cost of implementation and quality improvement. Concretely it achieves an improvement of 0.36 BLEU points and 0.4 METEOR points when translating noisy text.

However, oracle results show that there may be still margin for improvement on the heuristic strategies if a better ranking method for the hypotheses could be found. In the future we plan to study the behavior of bigger language models for open domain tasks (e.g. Google N-grams) and we will try to combine the heuristic and trained character-based phrase-tables in order to provide additional robustness to the proposed misspelling correction strategies.

Acknoweledgments

We would like to thank the anonymous reviewers for their insightful comments. This research has been partially funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 247762 (FAUST project, FP7-ICT-2009-4-247762) and by the Spanish Government (Buceador, TEC2009-14094-C04-01)

References

E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. 2008. Finding High-quality Content in Social Media. In *Proceedings of the International Conference on Web Search and Web Data Mining*, WSDM '08, pages 183–194, New York, NY, USA. ACM.

T. Aikawa, L. Schwartz, R. King, M. Corston-Oliver, and C. Lozano. 2007. Impact of Controlled Language on Translation Quality and Post-editing in a Statistical Machine Translation Environment. In *Proceedings of MT Summit XI*, pages 10–14.

A.T. Aw, M. Zhang, J. Xiao, and J. Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 33–40. Association for Computational Linguistics.

N. Bertoldi, R. Zens, M. Federico, and W. Shen. 2008. Efficient Speech Translation Through Confusion Network Decoding. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1696 –1705, nov.

N. Bertoldi, M. Cettolo, and M. Federico. 2010. Statistical Machine Translation of Texts with Misspelled Words. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 412–419, Stroudsburg, PA, USA. Association for Computational Linguistics.

E. Brill and R.C. Moore. 2000. An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 286–293, Stroudsburg, PA, USA. Association for Computational Linguistics.

C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proc. of the 7th Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada. ACL.

K.W. Church and W.A. Gale. 1991. Probability Scoring for Spelling Correction. *Statistics and Computing*, 1(2):93–103.

D. Contractor, T.A. Faruquie, and L.V. Subramaniam. 2010. Unsupervised cleansing of noisy text. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 189–196. Association for Computational Linguistics.

F.J. Damerau. 1964. A Technique for Computer Detection and Correction of Spelling Errors. *Communications of the ACM*, 7(3):171–176.

K. Dave, S. Lawrence, and D.M. Pennock. 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *Proceedings of the 12th International Conference on World Wide Web*, pages 519–528. ACM.

M. Denkowski and A. Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proc. of the 6th Workshop on Statistical Machine Translation*, pages 85–91. Association for Computational Linguistics.

S. Deorowicz and M.G. Ciura. 2005. Correcting Spelling Errors by Modelling their Causes. *International Journal of Applied Mathematics and Computer Science*, 15(2):275.

L. Dey and SK Haque. 2009. Studying the Effects of Noisy Text on Text Mining Applications. In *Proceedings of The 3rd Workshop on Analytics for Noisy Unstructured Text Data*, pages 107–114. ACM.

D. Fossati and B. Di Eugenio. 2008. I Saw TREE Trees in the Park: How to Correct Real-word Spelling Mistakes. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 896–901.

Q. Gao and S. Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio, June. ACL.

G. Hirst and A. Budanitsky. 2005. Correcting Real-word Spelling Errors by Restoring Lexical Cohesion. *Natural Language Engineering*, 11(01):87–111.

A. Islam and D. Inkpen. 2009. Real-word spelling correction using google web it 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1241–1249. Association for Computational Linguistics.

S. Jacquemont, F. Jacquenet, and M. Sebban. 2007. Correct your Text with Google. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 170–176. IEEE Computer Society.

C. Kobus, F. Yvon, and G. Damnati. 2008. Normalizing SMS: Are Two Metaphors Better than One? In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, pages 441–448. Association for Computational Linguistics.

P. Koehn and H. Hoang. 2007. Factored translation models. In *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June. ACL.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

G. Kothari, S. Negi, T.A. Faruquie, V.T. Chakaravarthy, and L.V. Subramaniam. 2009. SMS based Interface for FAQ Retrieval. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 852–860. Association for Computational Linguistics.

K. Kukich. 1992. Spelling Correction for the Telecommunications Network for the Deaf. *Commun. ACM*, 35:80–90, May.

L. Mangu, E. Brill, and A. Stolcke. 2000. Finding Consensus in Speech Recognition: Word Error Minimization and other Applications of Confusion Networks. *Computer Speech & Language*, 14(4):373 – 400.

Eric Mays, Fred J. Damerau, and Robert L. Mercer. 1991. Context based spelling correction. *Information Processing & Management*, 27(5):517 – 522.

R. Mitton. 1996. *English Spelling and the Computer*. Longman Group.

F.J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ll. Padró, M. Collado, S. Reese, M. Lloberes, and I. Castellón. 2010. Freeling 2.1: Five years of open-source language processing tools. In *Proc. of 7th Language Resources and Evaluation Conference (LREC 2010)*, La Valletta, MALTA, May. ELRA.

J. Pedler. 2007. *Computer Correction of Real-word Spelling Errors in Dyslexic Text*. Ph.D. thesis, Birkbeck, University of London.

L. Philips. 2000. The Double Metaphone Search Algorithm. *C/C++ Users J.*, 18:38–43, June.

D. Pighin, Ll. Màrquez, and Ll. Formiga. 2012. The faust corpus of adequacy assessments for real-world machine translation output. In *Proc. of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

A. Stolcke. 2002. SRILM-an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 901–904.

L.V. Subramaniam, S. Roy, T.A. Faruquie, and S. Negi. 2009. A Survey of Types of Text Noise and Techniques to Handle Noisy Text. In *Proceedings of The 3rd Workshop on Analytics for Noisy Unstructured Text Data*, AND '09, pages 115–122, New York, NY, USA. ACM.

K. Toutanova and R.C. Moore. 2002. Pronunciation Modeling for Improved Spelling Correction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 144–151. Association for Computational Linguistics.

FranÇois Yvon. 2010. Rewriting the orthography of sms messages. *Nat. Lang. Eng.*, 16(2):133–159, April.