Dealing with Input Noise in Statistical Machine Translation

Lluis Formiga[1] *Jose A. R. Fonollosa*[1]
(1) Universitat Politècnica de Catalunya
{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.

TITLE AND ABSTRACT IN ANOTHER LANGUAGE, SPANISH

Estudio de estrategias para tratar los errores ortográficos en la entrada de los sistemas de traducción automática estadística

Las palabras con errores ortográficos tienen un impacto directo en la calidad final obtenida por los sistemas de traducción automática estadística (TA) debido a que la entrada se vuelve ruidosa e impredecible. Este artículo presenta algunas estrategias de mejora a la hora de traducir textos de entrada con ruido del mundo real. Estas estrategias consisten en la adición de un paso de preproceso basado en un traductor a nivel de carácter de texto ruidoso a texto limpio. El uso de un traductor a nivel de carácter permite proporcionar diversas alternativas de ortografía en un formato de *lattice* como entrada del traductor bilingüe final. Por lo tanto, es el traductor final quien decide la mejor secuencia de palabras a traducir. Para esta tarea, las diferentes hipótesis se obtienen bajo suponiendo un modelo de distorsión del canal. En este trabajo presentamos los experimentos realizados con textos reales de entrada ruidosa y un sistema estándar de traducción auotmática estadística de inglés a español.

KEYWORDS: Noisy Text, Statistical Machine Translation, Social Media, Xat, SMS, Web2.0.

KEYWORDS IN SPANISH: Texto ruidoso, Traducción Automática Estadística, Medios Sociales, Chat, SMS, Web2.0.

Proceedings of COLING 2012: Posters, pages 319–328,
COLING 2012, Mumbai, December 2012.

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). On the other side, Bertoldi et al. (2010) presented a preliminary work focused on preserving all spelling alternatives to the input of MT system through Confusion Networks (CNs). However, this preliminary work was focused on an artificially generated noise that is not able to cover all the different properties of real-scenario weblog noise.

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.

2 Related work

Misspelling correction has been a recurrent issue to be resolved on NLP since its very first beginnings (Damerau, 1964). Good surveys of different types of noisy text and its related spell-correction programs can be found in Pedler (2007); Subramaniam et al. (2009); Kukich (1992) along with (Mitton, 1996).

According to Deorowicz and Ciura (2005), misspelling correction methods can be separated as *isolated-word error detection-correction* methods (Damerau, 1964; Philips, 2000; Toutanova and Moore, 2002), 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(Deorowicz and Ciura, 2005; Pedler, 2007; Jacquemont et al., 2007). Usually Noisy-Channel model is assumed for this task.

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). We did not consider context-dependent strategies due to their dependency to several language-specific analysis tools, which are beyond the scope our study.

3 Adaptive spelling correction based on character-based translation models

The strategy presented by Bertoldi et al. (2010) 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 novelty of

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
⋮	⋮	⋮	⋮	⋮	⋮

Table 1: Heuristic phrase table used for the spelling hypotheses generator (Moses decoder).

their work is the method employed to generate spelling alternatives, which it is a character-based decoder of heuristically defined CNs. Thus, 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 (Formiga and Fonollosa, 2012); 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.

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). 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 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. Therefore, 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, the noise of chat/SMS domains concerns higher level errors. Within these errors we can distinguish two types: *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.

Similar to Contractor et al. (2010), 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 (See details in Formiga et al. (2012)). In this experiments, we preprocessed the text to lowercase in order to overcome the casing problems, which are quite frequent under noisy scenarios. 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 (Agichtein et al., 2008; Subramaniam et al., 2009).

	Perplexity		WER	
Data	DEV	TEST	DEV	TEST
Original Source (wr)	835.713	891.55	–	–
Clean Source 0 (w0)	541.58	533.74	13.54%	16.33%
Clean Source 1 (w1)	575.35	660.34	8.61%	6.51%
Combined Clean Sources (w0.w1)	558.39	594.03	6.67%	6.35%

Table 2: Perplexity and WER obtained between original and cleaned data.

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].

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. Hence, there are three different test sets from this material: *i*) *Weblog Raw* (*wr*): The noisy weblog input, *ii*) *Weblog Clean*$_i$ (*w0* and *w1*): the cleaned version of the input text provided by different humans on the source side. Cleaned versions may differ due to the interpretation of the translators and *iii*) *Weblog Clean0.1* (*w0.w1*): the cleaned versions with mixed up criteria. In that case the cleaned versions are concatenated (making up a set of 3994 sentences). In order to perform the different optimization tasks, we have divided the noisy set in development (999 sentences) and test (998 sentences) sets.

We analyzed through some indicators the presence of noise within the weblog data sets following the work performed by Subramaniam et al. (2009). Concretely we measured 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).

Results are detailed on table 2. 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 all the test sets, 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 compared the different strategies presented in Sections 2 and 3. They are named *i*) Distance (Levenshtein distance plus a LM), *ii*) Confusion (Bertoldi et al., 2010), *iii*) Heuristic PT (heuristically defined phrase-table) and *iv*) GIZA PT (monolingual char-based MT). In the latter case we post-edited manually 8000 noisy sentences submitted to the same portal (Softissimo), so they were 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. It had 8 weights to be tuned (5 phrase-table model weights, language model, character penalty and distortion).

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

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→Spa translator required 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).

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 3: CER/WER/BLEU/METEOR scores obtained when cleaning the texts.

Results are detailed in table 3. 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.

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 4: BLEU scores obtained applying different misspelling MT strategies

The detailed results (BLEU) are shown in Table 4. A more detailed analysis might be found in Formiga and Fonollosa (2012)

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. Also we can observe slightly improvements on the clean sets: ≈ 0.16 BLEU 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 → I will go or I 'll go). In that sense the misspelling correction step also performs a revision of the tokenization carried out beforehand. 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 3. 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 → that is fun) as well as high-level errors (e.g. prove'em wrong → 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 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

Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. (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.

Aikawa, T., Schwartz, L., King, R., Corston-Oliver, M., and Lozano, C. (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.

Aw, A., Zhang, M., Xiao, J., and Su, J. (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.

Bertoldi, N., Cettolo, M., and Federico, M. (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.

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

Brill, E. and Moore, R. (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.

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

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

Denkowski, M. and Lavie, A. (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.

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

Dey, L. and Haque, S. (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.

Formiga, L. and Fonollosa, J. A. R. (2012). Correcting input noise in SMT as a char-based translation problem. In *TALP internal report*, UPC, Barcelona. `http://nlp.lsi.upc.edu/publications/papers/misspelling_techrep_oct2012.pdf`

Formiga, L., Henríquez Q., C. A., Hernández, A., Mariño, J. B., Monte, E., and Fonollosa, J. A. R. (2012). The talp-upc phrase-based translation systems for wmt12: Morphology simplification and domain adaptation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 275–282, Montréal, Canada. Association for Computational Linguistics.

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

Kobus, C., Yvon, F., and Damnati, G. (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.

Koehn, P. and Hoang, H. (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. ACL.

Kothari, G., Negi, S., Faruquie, T., Chakaravarthy, V., and Subramaniam, L. (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.

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

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

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

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

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

Pighin, D., Màrquez, L., and Formiga, L. (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. European Language Resources Association (ELRA).

Subramaniam, L., Roy, S., Faruquie, T., and Negi, S. (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.

Toutanova, K. and Moore, R. (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.

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