Holaaa!! writin like u talk is kewl but kinda hard 4 NLP

Maite Melero, Marta R. Costa-Jussà, Judith Domingo, Montse Marquina, Martí Quixal
Barcelona Media
Barcelona, Spain
maite.melero, marta.ruiz, judith.domingo, montse.marquina, marti.quixal@barcelonamedia.org

Abstract

We present work in progress aiming to build tools for the normalization of User-Generated Content (UGC). As we will see, the task requires the revisiting of the initial steps of NLP processing, since UGC (micro-blog, blog, and, generally, Web 2.0 user texts) presents a number of non-standard communicative and linguistic characteristics, and is in fact much closer to oral and colloquial language than to edited text. We present and characterize a corpus of UGC text in Spanish from three different sources: Twitter, consumer reviews and blogs. We motivate the need for UGC text normalization by analyzing the problems found when processing this type of text through a conventional language processing pipeline, particularly in the tasks of lemmatization and morphosyntactic tagging, and finally we propose a strategy for automatically normalizing UGC using a selector of correct forms on top of a pre-existing spell-checker.

Keywords: UGC, Social Media, text normalization, selection of correction proposals, language model combination

1. Motivation for this work

The Web 2.0 has become a channel where users exchange, explain or write about their lives and interests, give opinions and comment on other people's opinions, most of the times using a casual language with particular idiosyncrasies that make it much closer to oral language than to standard edited text. Opinion mining techniques, just to mention an example, are becoming an important source of information for market research. In order to mine the data or extract information from the Web 2.0 we need first to understand the contents in it. Shortened or misspelled words, which are very frequent in the Social Media informal style, increase the variability for the same concept. NLP techniques, which are used to analyze text and provide formal representations of surface data, have been typically developed to deal with standard language and may not yield the expected results on User Generated Content text. UGC in text form is a valuable resource that can be exploited for many purposes. The "massaging" of the input text so that it can be properly processed by standard NLP tools is often called Text Normalization in the literature.

[Kobus et al, 2008] present an interesting discussion on three different "metaphors" or ways of looking at SMS language, a type of text that has some features in common with UGC text. Each of these views motivates for a different approach to the normalization task. In the first approach, each input token is taken as a deviation of the correct word form, and normalization is thus viewed as a spell checking task. The second metaphor considers SMS language as a different language, and so normalization can be viewed as a machine translation task. Finally, it is possible to consider normalization as a speech recognition task because some people consider SMS as being closer to oral productions than to regular written texts. In fact, SMS spellings tend to be a closer approximation to the phonemic representation of a word than to its normative spelling. Finally, a fourth approach would be to completely forgo normalization by dealing with UGC peculiarities as a particular instance of domain adaptation. That is, instead of transforming the data so that it resembles the parser's training data, transform the parser's training data so that it resembles the input data. The strategy presented in [Foster, 2010] is a combination of both normalizing the input text and adapting the training models.

In this paper, we first present our findings on what characterizes UGC text in Spanish, based on a corpus study. We then explore the problems caused by UGC text to NLP tools performance, by comparing the results of parsing two versions of the same UGC text: as-is and manually corrected. Finally, we present an approach to text normalization that uses a language model-based automatic correction selector, built on top of a pre-existing spellchecker. As far as we know little work has been made to date on the subject for Spanish, with a few exceptions [Alonso, 2010].

2. Corpus-based characterization of UGC text in Spanish

As a reference corpus to study UGC related phenomena, we have collected a sample of texts in Spanish from the following sources: blogs (collected using Google Blog Search), hotel reservations (booking.com), consumer reviews in three different domains (ciao.es) and Twitter. The total size of the sample is 7583 sentences, or 192417 words. We have manually revised the corpus and corrected each deviation from standard language norms (sometimes referred to as errors, but not necessarily). Each deviation has been assigned one type among the following:

- **Capitalization**: The text is capitalized for emphasis or emotive purposes, or proper nouns are not capitalized: "y NO es broma" [*NOT kidding*] for "y no es broma" or "me recorro españa" [*I go around spain*] for "me recorro España".
- **Accentuation**: Graphical accents are omitted: "en numeros rojos" for "en números rojos" [*in the red*].
- **Punctuation**: Punctuation signs are omitted or

reduplicated; this includes also omission of blank spaces: "Te quiero!!!!!!!" for "¡Te quiero!" [*I love you!*], "aver" for "a ver" [*let's see*].

- **Informal Spelling**: All systematic shortcuts and character substitutions intentionally made by the user: "pq" for "porque" [*because*], "t kiere muxo" for "te quiere mucho" [*he loves you so much*].
- **Spelling errors**: All spelling errors not included in the previous categories, including conventional misspellings, such as "oie" for "oye" [*listen*] or “targetas” for “tarjetas” [*cards*]; typos, such as “diciendo” for “diciendo” [*saying*]; and intentional or unintentional reduplication of characters: as in “cooordenadas” for “coordenadas” [*coordinates*] , “alistarmeeeee” for “alistarme” [*join up*] or "frrrrrrío" for "frío" [cold].
- **Other errors** (lexical, syntactic): e.g., agreement errors or missing prepositions: "delante mi casa" for "delante de mi casa" [*in front of my house*], "mucho gente" for "mucha gente" [*lots of people*].

2.1 Corpus size

Table 1 shows the size of the corpus in terms of sentences and words, and the corresponding percentage of each source or domain. Ciao comprises texts from three different domains: car (61%), mobile operators (12%) and banking (26%). The last row shows the ratio between total number of words and number of different words appearing at least once (word types), which is smaller for Twitter and Booking, revealing a greater lexical variation in these types of text.

	Total	Twitter	Blog	Booking	Ciao
# Sents.	**7583**	20%	21%	2%	57%
# Words	**192417**	14%	23%	1%	62%
Ratio	**6.0**	3.1	6.0	3.4	7.6

Table 1: Corpus size in sentences and words, and ratio of word types.

2.2 Annotation rationale

The normalization of ill-formed text presupposes a definition of norm. The concept of norm may vary from one linguistic community to the other. For example, norm is reached by consensus in the English-speaking world, but it is dictated by a prescriptive institution for Spanish or French. In addition, within a particular community or corporation it can be further restricted with arbitrary goal-driven norms. From the perspective of NLP, if a word form contains deviations from the normalized standard, the system might fail to annotate it with the appropriate linguistic information.

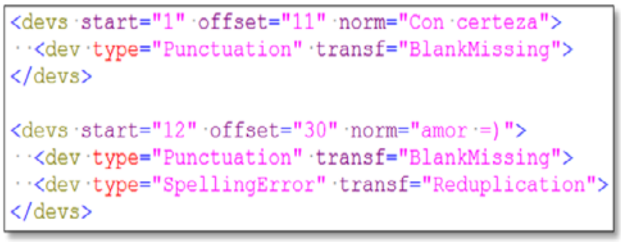
2.3 Annotation criteria

After an initial inspection of the corpora we decided to manually correct the texts using the following criteria:

1. Read and note deviating forms
2. Mark the span of the deviating form
3. Write the alternative normalized version of the text
4. Classify the deviating form according to the following types:
    a. Linguistic type: as described in Section 2.
    b. Transformation type: addition, omission, substitution, transposition and duplication.

The initial manual annotation was then mapped into an XML annotation scheme. This scheme is scalable and compatible with the Text Encoding Initiative (TEI) conventions. It is conceived as a stand-off annotation: instead of mixing the data with the metadata, the original text is preserved as-is while the annotation forms a separate layer, linked to the original text through offset indicators.

Figure 1 shows the annotated example segment coming from Twitter: “Concerteza amooor=)”, that ends up being normalized as “|Con| |certeza| |amor| |=)|”, in which the three original tokens, including a space, derive in four normalized tokens, including three spaces.



```
<devs start="1" offset="11" norm="Con certeza">
  <dev type="Punctuation" transf="BlankMissing">
</devs>

<devs start="12" offset="30" norm="amor =)">
  <dev type="Punctuation" transf="BlankMissing">
  <dev type="SpellingError" transf="Reduplication">
</devs>
```

Figure 1: XML-based stand-off annotation.

2.4 Description of UGC text

Overall, the rate of deviated input in our UGC corpus is quite high: over a fifth of the words (20.8%) contain some error or deviation. This rate varies according to type of text, going from 4.62% in more edited text, such as blog posts, to over 25% in Twitter and informal consumer reviews.

Table 2 shows the percentage of words which have been manually corrected with respect to the total of words in each corpus, classified according to the type of error or deviation. With minor exceptions, the frequency distribution of the deviation types does not exhibit significant variations across the different corpus and domains.

Due to their relative frequency, three types of deviations clearly stand out over the rest: spelling errors, capitalization and accentuation. Even though the “Spelling errors” class in our classification includes a certain amount of reduplications of characters with expressive or emotive purposes, most of the instances are “ordinary” orthographic errors, which, together with accentuation problems are well handled by conventional spellcheckers. This fact has undoubtedly motivated the solution we have chosen for dealing with UGC text as we explain in Section 4.

	Spelling errors (%)	Capitalization (%)	Accentuation (%)	Punctuation (%)	Informal Spelling (%)	Other (%)	**TOTAL ERRORS (%)**
TWITTER	8.11	7.29	6.25	1.77	1.52	0.68	**25.62**
BLOGS	2.64	1.38	0.41	0.12	0.01	0.06	**4.62**
BOOKING	3.71	1.71	8.71	0	0.57	1.56	**16.26**
CIAO-BANKING	4.98	12.34	9.68	0.35	0.14	0.08	**27.57**
CIAO-CARS	8.95	5.47	7.99	1.86	0.28	0.50	**25.05**
CIAO-MOBILE	5.70	10.84	9.78	1.47	0.68	0.93	**29.4**
TOTAL ACROSS DOMAINS	**6.31**	**6.30**	**6.08**	**1.11**	**0.57**	**0.43**	**20.8**

Table 2: Percentage of deviations according to its linguistic type, across domains

3. Processing UGC text

Our hypothesis is that the high frequency of deviations present in the text will have an impact on the performance of standard NLP tools. In a similar experiment, [Foster, 2010] detects problems with the handling of long coordinated sentences (mainly in the presence of erratic punctuation usage), domain-specific fixed expressions and unknown words.

In order to gauge the impact of deviations on the linguistic processing of UGC text we have processed the two versions of our corpus (original and manually corrected) using a conventional linguistic processing pipeline for Spanish [Rodríguez et al., 2010] and compared the outcome in terms of changes in the resulting annotation. The pipeline consists of state-of-the-art linguistic tools, integrated on a UIMA platform, which have not been adapted to this type of text. According to our analysis, the impact of normalizing deviated text varies between around 30% and 100% depending on type of error, task and domain.

3.1. Impact on lexical coverage

Not surprisingly, normalization of the input increases lexical coverage. Table 3 shows the percentage of words (both in terms of individual instances and word types) covered by the system's lexical resources, both in the original and in the manually normalized version. These values are notably lower for Twitter than for the rest of the sources.

The increase in coverage of the normalized version is shown between brackets. This increase turns out to be more evident in the comparison of word types than in the comparison of word instances, perhaps as a side effect of normalization of deviated forms also decreasing the number of hapax (words that appear only once).

		Original (%)	Normalized (%)
TWITTER	Inst.	81.3	83.7 (+2.4)
	Type	65.6	68.8 (+3.2)
BLOG	Inst.	95.5	96.3 (+0.8)
	Type	86.6	89 (+2.4)
BOOKING	Inst.	97.4	98.9 (+1.5)
	Type	92	96.2 (+4.2)
CIAO	Inst.	95.4	96.8 (+1.4)
	Type	80.4	85.4 (+5)

Table 3: Percentage of word coverage (instances and types) in both the original and the corrected versions.

3.2. Impact on the performance of three basic NLP processing tasks

In this work we have focused on the effect of normalization on three basic NLP processing tasks: (i) lemmatization, (ii) part-of-speech tagging (short-tag or syntactic category), and (iii) assignment of morphosyntactic features (gender, number, tense...) These tasks are at the root of more complex or higher level processing tasks, and errors at this level are likely to spread upwards and affect other tasks such as constituent analysis, dependency relations, NERC, etc.

Table 4 shows the percentage of words for which a change in the resulting analysis is found after normalization.

	Lemmatization (%)	PoS tagging (%)	Morph. Features (%)
BLOGS	94.6	43.8	62.6
CIAO-BOOKING	88.6	43.7	65.5
TWITTER	85.4	49.8	64.5
TOTAL CORPUS	**89.55**	**45.80**	**64.19**

Table 4: Percentage of deviating words incorrectly analyzed and tagged, by domain.

In general, results are fairly uniform across the three domains. We observe that assignment of part-of-speech label is generally quite robust to deviation, since only less than half of the deviated words change their PoS assignment after having been corrected or normalized. On the other hand, lemmatization is very sensible to the presence of error. As a matter of fact, our lemmatizer has been unable to assign a proper lemma practically in 90% of the instances of deviated or erroneous words.

Table 5 presents the same information broken down by deviation type, for the four most frequent types of deviation.

	Lemma-tization (%)	PoS tagging (%)	Morph. Features (%)
Capitalization	68.32	29.49	56.32
Accentuation	97.83	65.15	75.93
Spelling errors	97.87	56.01	71.12
Punctuation	99.83	46.09	59.09

Table 5: Percentage of deviating words incorrectly analyzed and tagged, by type of error.

Capitalization turns out to be the less detrimental across these basic tasks, while accentuation is particularly harmful, even for a robust task such as PoS tagging.

On the whole, we have seen that PoS tagging of UGC is affected by error in 50% of cases, a little more in the case of assignment of morphosyntactic features. As for lemmatization of UGC, this task is likely to be inaccurate for most deviated words. Errors of lemmatization increase the variability for the same concept and thus are likely to affect most semantic related tasks.

4. Building a Normalizer on top of an existing Spellchecker

The large presence of deviated forms in UGC text and its costly impact on the performance of NLP tools has convinced us of the necessity of searching for a solution that addresses the problem. As discussed in Section 1, we find two divergent approaches to deal with this issue in the literature: either transform the input text (i.e. normalize it) or transform the tools. While the second option (in part followed by Foster, 2010) is feasible mostly for statistically trained tools, the first option should work for any tool, statistical and rule-based. Of the different approaches to normalization discussed in Section 1, we have chosen to view normalization as a spellchecking task, particularly motivated by the high rate of "typical" orthographic errors (including accentuation) in UGC text.

In our case, we have built our normalizing tool on top of the Spanish version of COTiG, a spell and grammar checker architecture first developed for Catalan [Quixal et al. 2008]. We have used the annotated corpus described in Section 2 as a Gold Standard or reference corpus for evaluation.

A key difference between a regular spellchecker and a normalizer is interactivity with an end-user. The lack of interactivity in the normalization task has an important implication: A specific strategy has to be put in place in order to rank possible correction candidates and decisively choose the best one over the whole set.

A second aspect concerns false positives. If overcorrecting may be annoying for the user of a spell checker, overnormalizing, i.e. introducing unwanted changes to the original text, can be invalidating for a normalizer.

Finally, typical UGC phenomena, such as informal spellings or emoticons. may not be appropriately dealt with by using standard spellchecking procedures.

To sum up, the modifications to the base correction engine involve:

a) Inclusion of a dedicated module that ranks the list of correction candidates and selects the highest ranked one. This module is described in more detail in section 5.

b) Domain adaptation in order to reduce number of unknown words, and therefore, the number of false positives, by using domain dictionaries created out of frequency vocabulary lists extracted from in-domain corpora and other linguistic resources.

c) Inclusion of specific treatments for typical UGC phenomena, such as "Informal spellings", which cannot be dealt with by using standard editing distance algorithms (e.g. letter substitution of ch by x, of qu by k, word substitution of por by x, que by q, etc.), reduplication of characters (holaaaaaaaaa, dormirrrr, ...), emoticons, etc.

5. Selection of the right candidate using language models

N-gram models have been used for the detection and correction of misspellings in isolated words since the late eighties (Kukich 1992). Gale and Church (1990) demonstrate the potential of word bigrams to improve the accuracy of isolated word correction, and Mays et al. (1991) using trigram models obtain 76% accuracy in detection and 74% accuracy in correction.

The output of the spellchecker typically consists of an unordered list of correction candidates obtained through the application of its own correction algorithms, which include editing-distance criteria.

```
<devs begin="318" end="322" original="coxes">
  <proposals>
    <proposal id="1">boxes</proposal>
    <proposal id="2">comes</proposal>
    <proposal id="3">coses</proposal>
    <proposal id="4">coches</proposal>
    <proposal id="5">coxas</proposal>
    <proposal id="6">coges</proposal>
    <proposal id="7">corres</proposal>
    <proposal id="8">coxis</proposal>
    <proposal id="9">coles</proposal>
    <proposal id="10">boches</proposal>
    <proposal id="11">coces</proposal>
  </proposals>
 </devs>
```

Figure 2: Output of the normalizer in XML format containing a list of correction candidates

In order to select the most probable correction among a set of candidates proposed by the underlying spellchecking engine we have experimented with the use of different trigram models, based on the same source corpus, but each conveying a different degree of information:

- True-case form model (TC). This model has been trained on the original unmodified text, where upper-case and lower case instances of the same form are different words. It is the least general and the most informative.

- Lower-case form model (LC). This model has been trained on the lowered-case version of the original corpus. Upper-case and lower case versions of the same form are now the same word.
- Lemma model (Lemma). This model has been trained on the lemmatized version of the original corpus, where each inflected form has been substituted by its root or lemma. Plural and singular versions of the same noun are now the same word; the same happens with variations in person, tense of number of verbal forms.
- Part-of-Speech model (PoS). This model has been trained on the PoS tags version of the corpus. Tags are Parole style part-of-speech long tags, which for each word include its syntactic category and morphosyntactic features (e.g. AQ0CP0, NCFP000, VSIC3P0). This model is the least informative and the most general.

5.1 Building the models

The corpus used to build the models is an 18 Million word corpus collected from the web, that comprises texts from the same domains and genres included in the reference corpus, namely: banking, cars, mobile, twitter and blogs. Two thirds of these texts are (unrevised) user generated content while one third comes from more edited sources. We extracted a development set out of the edited portion of the main corpus, trying to cover all domains in a balanced fashion. This development set of around 100K words, is used in the optimization phase.

In order to obtain the lemma and part-of-speech information we have processed the training and the development corpora using our in-house linguistic pipeline. We have then built the 4 trigram models with the IRSTLM toolkit (Federico and Bertoldi, 2006), using the "modified shift-beta" as smoothing method, also known as "improved Kneser-Ney smoothing".

5.2 Querying the models

At run time we query the models using a sliding window over the proposed correction of at maximum 5 words. For the example above in Figure 2, 11 different 5-word strings are generated, one for each different candidate surrounded by its immediate context (between brackets).

Original string: (..), *aunque[mire otros coxes de la] misma categoria, (..)*
Candidate strings:

S_1= *mire otros boxes de la*
S_2= *mire otros comes de la*
S_3= *mire otros coses de la*
S_4= *mire otros coches de la*
S_5= *mire otros coxas de la*
S_6= *mire otros coges de la*
S_7= *mire otros corres de la*
S_8= *mire otros coxis de la*
S_9= *mire otros coles de la*
S_{10}= *mire otros boches de la*
S_{11}= *mire otros coces de la*

For each candidate strings S_i, we generate 4 parallel strings:

- S_{TC} with all words in truecase: *mire otros boxes de la*
- S_{LC} with all words in lowercase: *mire otros boxes de la*
- S_{Lemma} with lemmas only: *mirar otro box de la*
- S_{PoS} with PoS only: VMSP3S0 DI3MP0 NCMP000 SPS00 DA3FS0

Being C(M,S) the cost (i.e. logarithm of the probability) of string S according to model M, we query each of the four models for the cost value of each one of the strings, e.g.: C(TC, S_{TC})=cost of the truecase string computed against the truecase model, etc. and get four cost values: C(TC, S_{TC}), C(LC, S_{LC}), C(Lemma, S_{LEMMA}) and C(PoS, S_{PoS}).

5.3 Combining the models

The aim of building the four models is to evaluate which model is more discriminative at ranking the different candidates and also to experiment with different combinations of the models:

C(total,S)= λ_0C(TC, S_{TC}) + λ_1 C(LC, S_{LC}) + λ_2 C(Lemma, S_{lemma}) + λ_3 C(PoS, S_{PoS})

The combination weights have been empirically adjusted by following the standard procedure of the language modeling interpolation. This standard procedure follows the minimum cost criterion. The perplexity for each of the 4-language models is computed over its corresponding development set (see Table 6 for statistics). Therefore we have to previously construct a lowercase version of the development set, a lemma version and a PoS version. The language model weight optimization was sentence-by-sentence, but one set of language model weights was determined for the whole development set using the compute-best-mix function of the SRILM toolkit (Stolcke, 2002).

		TC	LC	Lemma	PoS
Training	Sents	655803			
	Words	18839039			
	Vocab	480047	422864	449587	158
Dev	Sents	5347			
	Words	108189			
	Vocab	12542	11719	11019	120

Table 6: Training and development corpus statistics.

As shown in Table 6, while the number of words is obviously the same for the 4 training sets –and the 4 development sets-, the size of the vocabulary varies a lot, being smallest in the case of the PoS version.

Perplexity of each language model and their combination is shown in Table 7. The combination allowed a reduction of 10% respect to the smallest perplexity of the individual values. Table 7 also shows the out of vocabulary (OOV) words for each language model approach. Notice that perplexity increases with the number of "out of vocabulary" words. The PoS language model is the one that offers the lowest perplexity and the lowest number of "out of vocabulary" words.

	Perplexity	OOV
Truecase	299.483	11352
Lowercase	186.677	8434
Lemmas	216.817	11300
PoS	12.202	0
Combination	10.954	0

Table 7: Perplexity and Out of Vocabulary (OOV) words values over the development set for each language model (words in truecase, lowercase, lemmas and PoS) and the combination of all.

The resulting optimization yielded the following set of weights:

$C(total,S)= 0.00316558*C(TC, S_{TC}) + 0.104756 *C(LC, S_{LC}) + 0.0797074* C(Lemma, S_{lemma}) + 0.812371* C(PoS, S_{PoS})$

5.4 Evaluation of the selection of the correction candidate

To evaluate the module in charge of ranking the correction candidates we have run the normalizer on the reference corpus described in section 2. There is a total of 2463 instances where the spellchecker correctly detects an error and the reference correction is within the set of candidates proposed by the spellchecker. We have tested each of the 4 models individually and 2 model combinations: one in which each model has the same weigh and the optimized combination described in section 5.3. As a baseline we have run the normalizer without the selector module, i.e. keeping the first candidate proposed by the underlying spellchecker.
The results, in terms of percentage of instances where the reference correction was ranked first, are shown below in Tabl**e** 8.

Model	Precision
LC	86
TC	85.7
λ =0.25	81.9
Optimized λ	73.6
Lemma	72.3
PoS	64.7
Baseline	51.3

Table 8: Precision values for each model, their combination and baseline

The more discriminating model turns out to be the LC model, build on the lower-cased version of the corpus, with a precision of 86%, well over the baseline. The results also show that our simple optimization strategy was not adequate, and in fact the weighted combination presented in Section 5.3 assigns a disproportional importance to the least performing model (PoS), probably due to its low perplexity values. Since our models deal with different vocabularies, we were probably trying to merge apples with oranges.
A more promising procedure seems to be to optimize the weights directly in the final task, for example using algorithms such as Simplex (Spall, 1992) or SPSA (Spall 1998) to find the right combination of weights that minimizes the number of errors in the correction task. This is the direction that we are taking at the time of writing this paper.

6. Conclusion

In this paper, we have analyzed a Spanish corpus of UGC, and the impact that deviated text has on standard NLP processing tools. We have seen that UGC text presents some particular features that set it apart from standard text: on one hand, it contains specific phenomena that add expressivity, such as emoticons, informal spellings, non-standard capitalization and reduplication; and on the other hand, the rate of typical orthographic errors is much higher than in more edited types of text. We have also observed that UGC text has a negative impact on the performance of NLP tools, as measured in three basic tasks. To address the problem, we have used a conventional spellchecker modified in the following way: we have adapted it to domain by enriching its lexical base; we have added a limited set of rules to deal with specific phenomena, such as reduplication or informal spellings; and, finally, we have built a module to automatically select the best correction candidate. To build this module we have trained four language models on a big in-domain corpus, each model containing a different degree of information in a trade off with its generalization capabilities. The lowercase model has turned out to be the most predictive, with a reasonable precision value of 86%. However, our experiment has failed to produce an optimized combination of the four models. For this reason, we are currently in the process of attempting a different line of research, namely optimizing the model combination directly in the final task.

7. Acknowledgements

The research leading to these results has received funding from the CDTI through the CENIT project Social Media and from the Spanish Ministry of Economy and Competivity through the Juan de la Cierva fellowship program.

8. References

Alonso, L. (2010) Insights lingüísticos relativos a la normalización léxica de contenidos generados por usuarios. In Revista Subjetividad y Procesos Cognitivos. Buenos Aires.

Federico M. and Bertoldi N. (2006) How Many Bits Are Needed To Store Probabilities for Phrase-Based Translation?, In Proc. of the Workshop on Statistical Machine Transla- tion. pp. 94-101, NAACL, New York City, NY.

Foster, J. (2010) "cba to check the spelling" Investigating Parser Performance on Discussion Forum Posts. Proceedings of Human Language Technologies: 2010 Annual Conference of the North American Chapter of the ACL, , pp. 381--384, Los Angeles, CA

Foster, J., Cetinoglu, O., Wagner, J. and van Genabith, J. (2011) Comparing the Use of Edited and Unedited Test in Parser Self-Training. In Proceedings of IWPT, Dublin, Ireland.

Gale, W. A., and Church, K. W. (1990) Estimation procedures for language context: Poor estimates are worse than none. In Proceedings of Compstat-90, Dubrovnik, Yugoslavia. Springer-Verlag, New York, 69—74.

Kobus, C., Yvon, F., and Damnati, G. (2008). Normalizing SMS: are two metaphors better than one? In Proc. of the 22nd Int. Conf. on Computational Linguistics, pp. 441–448.Manchester.

Kukich, K. (1992) Techniques for Automatically Correcting Words in Text. ACM Comput. Surv. 24(4): 377-439 (1992).

Mays, E. Fred J. Damerau, and Robert L. Mercer. (1991). Context based spelling correction. Inf. Process. Manage. 27, 5 (September 1991), 517-522.

Quixal, M., Badia, T., Benavent, F., Boullosa, J. R., Domingo, J., Grau, B., Massó, G., Valentín, O. (2008) ”User-Centred Design of Error Correction Tools”. In Proceedings of the Sixth International Language Resources and Evaluation (LREC’08) Marrakech, Morocco.

Riseman, Edward M. and Hanson, Allen R., "A contextual post-processing system for error correction using binary n-grams," IEEE Trans Computers, vol. C-23, no. 5, pp. 480-493, May 1974.

Rodríguez,C.; Banchs,R.; Codina,J.; Grivolla, J. (2010) “COMETA: Semantic exploration of customer reviews to extract valuable information for business intelligence". Barcelona Media Technical Report, BM 2010-01.

Spall, C. (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation,” IEEE Trans. Automat. Control , vol. 37, pp. 332– 341.

Spall, C. (1998) “An overview of the simultaneous perturbation method for efficient optimization,” Johns Hopkins APL Technical Digest , vol. 19, no. 4, pp. 482–492.

Stolcke, A. (2002), SRILM -- An Extensible Language Modeling Toolkit. Proc. Intl. Conf. on Spoken Language Processing, vol. 2, pp. 901-904, Denver.

Tsao, Y. C. (1990, September). A Lexical Study of Sentences Typed by Hearing-Impaired TDD Users. Proceedings of the 13th International Symposium on Human Factors in Telecommunications, Turin, Italy.

VanBerkel, Brigitte and DeSmedt, Koenraad, "Triphone analysis: a combined method for the correction of orthographical and typographical errors," in Second Conference in Applied Natural Language Processing, Austin, Texas , pp. 77-83, American Association of Computational Linguistics, 1988.

Yannakoudakis, E.J. and Fawthrop, D., "The rules of spelling errors," Information Processing and Management, vol. 19, no. 2, pp. 87-99, 1983.