Elhuyar at TweetNorm 2013

Xabier Saralegi Urizar
Elhuyar Fundazioa
Zelai Haundi 3, 20170 Usurbil
x.saralegi@elhuyar.com

Iñaki San Vicente Roncal
Elhuyar Fundazioa
Zelai Haundi 3, 20170 Usurbil
i.sanvicente@elhuyar.com

Resumen: Este artículo presenta el sistema desarrollado por Elhuyar para la campaña de evaluación Tweet-Norm, que consiste en normalizar tuits en español a lenguaje estándar. La normalización abarca únicamente una serie de palabras fuera de vocabulario (OOV), previamente identificadas por la organización del taller. El sistema desarrollado utiliza una estrategia compuesta por dos pasos. Primero, para cada palabra OOV se generan posibles candidatos de correción. Para ello se han implementado diversos métodos que tratan de corregir diferentes tipos de errores: extensión de abreviaciones comunes, detección de coloquialismos, corrección de carácteres repetidos, normalización de interjecciones, y corrección de errores ortográficos mediante medidas de distancia de edición. En el segundo paso el candidato correcto es seleccionado utilizando un modelo de lenguaje entrenado sobre un corpus de español correcto. El sistema obtuvo un 68,3% de precisión sobre el corpus de desarrollo, y un 63,6% sobre el corpus de test, siendo el 4º sistema de la campaña de evaluación.
Palabras clave: Normalización de microtexto

Abstract: This paper presents the system developed by Elhuyar for the Tweet-Norm evaluation campaign which consists of normalizing Spanish tweets to standard language. The normalization covers only the correction of certain Out Of Vocabulary (OOV) words, previously identified by the organizers. The developed system follows a two step strategy. First, candidates for each OOV word are generated by means of various methods dealing with the different error-sources: extension of usual abbreviations, correction of colloquial forms, correction of replication of characters, normalization of interjections, and correction of spelling errors by means of edit-distance metrics. Next, the correct candidates are selected using a language model trained on correct Spanish text corpora. The system obtained a 68.3% accuracy on the development set, and 63.36% on the test set, being the 4th ranked system on the evaluation campaign.
Keywords: Microtext normalization

1 Introduction

Social media and specially Twitter have become a valuable asset for information extraction purposes. Twitter falls into the category of "microtext". As such, tweets present some characteristics which limit the straight application of natural language processing techniques: non standard orthography, colloquial expressions, abbreviations... So, converting Twitter messages to standard language is an essential step before applying any linguistic processing.

This paper presents the system developed by Elhuyar for the TweetNorm task, a task which consists of normalizing Spanish tweets. The normalization just covers the correction of certain OOV words. After tagging the reference using FreeLing (Padró et al., 2010), those words without analysis are regarded as OOV. The OOV list was provided by the organizers. Real-word errors are not treated in this task, that is, cases where a word is misspelled but the misspelled form also exists in the dictionary (e.g., *'té' -tea-* and *'te' -to you-*).

The developed system follows a two step

strategy. First, candidates for each problematic word are generated by means of various methods dealing with the different error-sources: extension of usual abbreviations, correction of colloquial forms, correction of replication of characters, normalization of interjections, and correction of orthographical errors by means of edit-distance metrics. The second step selects the correct candidate, by comparing the adequacy of each candidate against a language model trained from standard Spanish text corpora. The EFE news corpus and the Spanish Wikipedia were used for such purposes. The system obtained a 68.3% accuracy on the development set, and 63.6% on the test set, being the 4th ranked system on the evaluation campaign.

2 Related Work

In the last few years many researchers have turned their efforts to microblogging sites such as Twitter. However, the special characteristics of the language of Twitter require a special treatment when analyzing the messages. A special syntax (RT, @user, #tag,...), emoticons, ungrammatical sentences, vocabulary variations and other phenomena lead to a drop in the performance of traditional NLP tools (Foster et al., 2011; Liu et al., 2011).

To solve this problem, many authors have proposed a normalization of the text, as a pre-process of any analysis, reporting an improvement in the results. Han and Baldwin (Han and Baldwin, 2011) use morphophonemic similarity to match variations with their standard vocabulary words, although only 1:1 equivalences are treated, e.g., *'imo = in my opinion'* would not be identified. Instead, they use an Internet slang dictionary to translate some of those expressions and acronyms. Liu et al. (Liu, Weng, and Jiang, 2012) propose combining three strategies, including letter transformation, "priming" effect, and misspelling corrections.

3 Our System

The system performs the normalization process of tweets in two steps (see Figure 1). In a first step several methods are applied for generating candidates for the OOV words. In the next step a single candidate is selected for each OOV word by using language models.

Two data-sets were provided by the organizers of the Tweet-Norm event. One development-set C_{dev} composed of 500 tweets, and one test-set C_{test} composed of 600 tweets which was used only for evaluation purposes.

3.1 Generation of candidates

Some of these methods use reference lexicons for generating candidates. A reference lexicon of correct forms D_r was built by joining the FreeLing's dictionary forms and forms extracted from the EFE news corpus (146M words) and Spanish Wikipedia corpus (41M words), which theoretically include correctly written texts. A minimum frequency threshold was established in order to avoid possible typos, because several of them were found in both EFE and Wikipedia (e.g., *'tambien'*). A disadvantage of using these corpora is that they are focused on formal registers while the register of twitter is more informal. However, it is a difficult task to compile a corpus for informal register without including many wrongly written texts. So we sacrificed register adaption in benefit of correctness.

Colloquial vocabulary (COL)

We created a list of colloquial vocabulary (e.g., *'colegui', 'caseto', 'bastorro'*) by collecting words from two sources: *"Diccionario de jerga y expresiones coloquiales"*[1] dictionary and *www.diccionariojerga.com*, a crowdsourcing web including colloquial vocabulary edited by users. A different word corresponding to the correct form was inserted if necessary, otherwise the word itself is inserted as correct form. This list $L_c = \{(c_i, c'_i)\}$ contains 1088 entries.

The method based on this list is simple, if an OOV word c_i is included in the list the corresponding correct form c'_i is generated as a candidate.

Abbreviations (ABV)

A list containing the most used abbreviations (e.g.,*'mñn'*→*'mañana'*) and contractions (forms that join more than one word, e.g.,*'porfa'*→*'por_favor'*) in Twitter was created. First, the most frequent OOV words of a Twitter corpus (309,276 tweets, 4M tokens) were extracted, and the top 1,500 candidates ($freq(abv_i) > 25$) were analyzed, looking for abbreviations and contractions. Their corresponding correct forms were established by

[1] http://www.ual.es/EQUAL-ARENA/Documentos/coloquio.pdf

hand. As a result, 188 abbreviations were included in the list $L_{abv} = \{(abv_i, abv'_i)\}$. As with the previous method, for each OOV abv_i included in the list its standard form abv'_i is proposed as a correct candidate.

Interjections (INTJ)

Regular expressions were created for matching and normalizing the most common interjections and their variations (e.g., *'jeje'*, *'puf'*), identified in the development corpus C_{dev}.

Repeated letters (REP)

Repeated letters are removed from an OOV word if it does not appear in the reference lexicon D_r. Then if the modified form appears in D_r (e.g., *'calooor'*→*'calor'*) it is included as candidate.

Proper Nouns (PN)

A list of usual proper nouns was built from the Wikipedia corpus. Words in uppercase w_{uc} with a minimum frequency ($freq(w_{uc}) > 100$) and whose frequency is higher than that of their form in lowercase ($(freq(w_{uc}) > freq(w_{lc}))$) are taken as secure proper nouns. 6,492 words were collected in this manner.

If an OOV word w appears in a list of usual proper nouns and its first character is in lowercase then it is put in uppercase (e.g., *'betis'*→*'Betis'*).

Uppercase (UC)

If all characters of an OOV w_{uc} word are in uppercase the following rules are applied:

- If w_{uc} appears as it is in D_r, w_{uc} is proposed as candidate (e.g., *'IVA'*→*'IVA'*).
- If w_{uc} is included in D_r in lowercase $w_{lc} = lc(w_{uc})$, then w_{lc} is proposed as candidate (e.g., *IMPORTANTE*→*'importante'*).
- If w_{uc} is included in D_r with the first character in uppercase $w'_{uc} = ucfirst(w_{uc})$, then w'_{uc} is proposed as candidate (e.g., *'MADRID'*→*'Madrid'*).

Spelling errors (COG)

String similarity measures are useful for detecting correct forms of misspelled words. If the string similarity between an OOV word w and a correctly written word w' exceeds a certain threshold we can take w' as a correct candidate. We apply edit distance as follows: first, a set of transliteration rules are applied to both words ($trans(w)$ and $trans(w')$) in order to normalize some characters (e.g., $b = v$, $ki = qui$, $ke = que$ ...). Then, Longest Common Subsequence Ratio (LCSR) is calculated between $trans(w)$ and $trans(w')$. In order to reduce the computational cost of the process, LCSR is only computed for those words in our lexicon D_r that share the first character (except for h) with w and have a similar length (±20%). LCSR gives a score between 0 (minimum similarity) and 1 (maximum similarity). Those forms that reach a score greater than 0.84 are taken as candidates.

3.2 Selection of correct candidates

A tweet $t = \{f_0, ..., f_i, ...f_n\}$ can contain more than one OOV word, and each OOV word f_i can have several candidates $\{f_{i0}, ..., f_{ij}, ..., f_{im}\}$ after applying the above-mentioned methods (see Figure 1). Thus, a disambiguation process must be applied in order to obtain a single correct candidate for each OOV word. For that aim we use language models. The system selects for each tweet, the combination of candidates that best fits the language model, that is, the combination which maximizes the log probability of the sequence of words.

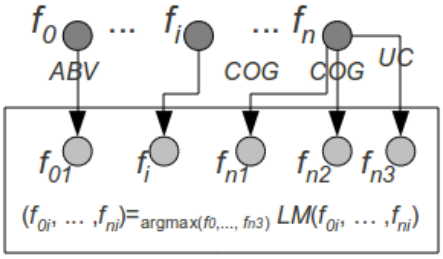

Figure 1: The diagram shows the two steps of the normalization process.

SRILM toolkit (Stolcke, 2002) was used for training and applying the language model. For the training process the EFE news corpus and the Spanish Wikipedia corpus were used. As mentioned in section 3.1, we chose those sources in order to guarantee maximal language correctness.

4 Results

Table 1 shows the results for the experiments done on the 500 tweets of the development collection C_{dev}, depending on the different

treatments and disambiguating by using an unigram language model trained on Wikipedia and EFE corpora. The baseline consists of selecting the OOV itself as correct candidate.

All the methods proposed provide an improvement over the baseline except for the UC method (see Table 1). The degree of improvement provided by each method varies depending on the frequency of the error-type treated by the method and the performance of the method itself. Thus, the string similarity based method COG provides the highest improvement (76.93% over the baseline), which means that the presence of typos is high and that the performance of the method is good. Both REP and ABV methods offer an improvement around 40% over the baseline. The treatment of interjections (INTJ) is also important, providing an improvement of 25% over the baseline. The error-types treated by the COL and UC are very scarce (14 and 5 respectively on C_{dev},). In the first case, although the methods perform well, the improvement is small. In the case of UC, most of the cases (4 out of 5) concatenate various error-types, and our system can not deal with error concatenations, leading to a performance decrease. Nevertheless, the method does provide an improvement when it is used combined with a bigram or a trigram LM, and thus, we include it in the all configuration. Error-types treated by PN are a bit more frequent ($\simeq$ 40 in C_{dev}). Although the method is quite precise ($P \simeq$ 80%) it lacks coverage ($R \simeq$ 60%). Among all method combinations the best accuracy was achieved when all of them were combined (ALL). So, we conclude that the LM manages properly the candidates provided by all the methods.

We performed further experiments with different orders of n-grams and different configurations of corpora, using in all cases the ALL configuration. According to the results (table 2), when larger orders of n-grams are used higher accuracies are obtained. This improvement is significant between 1-gram and 2-gram models. There is no improvement when using larger orders of n-grams. As for the corpora used, combining Wikipedia and EFE corpora provides the best performance. So it seems that they complement each other. Thus, evaluation over the test-set C_{text} was carried out using the bigram LM trained over the joint corpus between EFE and Wikipedia (See fifth column in Table 2).

	Acc. on the Devel. set	**Improvement over Baseline**
Baseline	23.28	-
Baseline+COL	24.2	3.95%
Baseline+ABV	32.16	38.14%
Baseline+INTJ	29.1	25%
Baseline+REP	34	46.05%
Baseline+PN	24.81	6.57%
Baseline+UC	23.12	-0.69%
Baseline+COG	41.19	76.93%
ALL	66.16	184.19%

Table 1: Accuracies for the candidate generation methods. Last column shows the improvement the method achieves over the baseline.

	Development			**Test**		
	unigr.	bigr.	trigr.	unigr.	bigr.	trigr.
EFE	64.93	66.62	66.62	-	-	-
Wikipedia	65.54	67.69	67.69	-	-	-
EFE + Wikipedia	66.16	**68.30**	67.69	-	63.60	-

Table 2: Accuracies for the different language models experiments, using the ALL configuration for the generation of candidates.

Error analysis

We performed error analysis over the OOV words not treated correctly by our best system for the 500 tweets of the development collection C_{dev}. Following, we explain the main problems detected in our system:

- Concatenation of errors: Generation methods are not combined between each other because LM is not capable of properly managing the noise created (e.g., *'SOI'→'SOY'→'soy'*, *'cumpleee'→'cumple'→'cumpleaños'*).
- Abbreviations and contractions: The abbreviation and contractions not included in our list are not properly normalized (e.g.,*'cmun'→'común'*, *'deacuerdo'→'de_acuerdo'*). LCSR based method is not capable of finding the correct form for the case of abbreviations either, because the distance is very high. If the threshold is decreased too much noise is created.

- Lack of domain adaptation: LM is trained from corpora corresponding to news and Wikipedia domains where informal register is not included. Because of that there are some colloquial expressions (e.g., *'maricón', 'bonico', 'comidita'*) and proper nouns (e.g., *'Pedrete', 'Fanegol'*) that are not included in our reference lexicon D_r and which are not properly disambiguated.
- Keyboard typos: Some errors correspond to key confusion at writing time. In some cases LCSR is not reached. (e.g., *'pa'→'la', 'tenho'→'tengo'*).

5 Conclusions

This paper presents a system for normalizing tweets written in Spanish. The system first generates a number of possible correction candidates for OOV words and then selects the candidate that better matches a language model trained over corpora of standard Spanish. Our system achieved the 4th rank among thirteen contestants in the tweet-Norm evaluation campaign. We consider this a satisfactory performance taking into account that, aside from the best system, the next four contestants are quite close to each other. Furthermore, our error analysis has shown that we still have room for improvement.

Edit distance must be adapted to better deal with abbreviations, contractions and keyboard errors. An alternative to improve that aspect could be to use a more complex strategy based on finite state toolkits such as Foma (Hulden, 2009).

On the other hand, we apply the different candidate generation methods in parallel, they are not combined in any way. This leads to a poor performance when an OOV has several errors concatenated. Therefore, we should explore possible method combinations, avoiding at the same time to generate too much noise, because the LMs would lose disambiguation capacity. In addition, we could experiment with larger LMs, and also LMs that are more focused on informal register.

Acknowledgments

This work has been partially founded by the Industry Department of the Basque Government under grant IE11-305 (KnowTOUR project).

References

Foster, Jennifer, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. #hardtoparse: POS tagging and parsing the twitterverse. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Han, Bo and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the ACL-HLT*, pages 368–378, June.

Hulden, Mans. 2009. Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, EACL '09, pages 29–32.

Liu, Fei, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the ACL*, pages 1035–1044, Jeju Island, Korea, July.

Liu, X., S. Zhang, F. Wei, and M. Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the ACL-HLT*.

Padró, Lluís, Miquel Collado, Samuel Reese, Marina Lloberes, and Irene Castellón. 2010. Freeling 2.1: Five years of open-source language processing tools. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*.

Stolcke, Andreas. 2002. Srilm - an extensible language modeling toolkit. In *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002*, pages 901–904.