A Method to Lexical Normalisation of Tweets*

Un método de normalización léxica de tweets

Pablo Gamallo y **Marcos Garcia**
CITIUS
Univ. de Santiago de Comp.
pablo.gamallo@usc.es

José Ramom Pichel
Imaxin Software
jramompichel@imaxin.com

Resumen: Este artículo describe una estrategia de normalización léxica de palabras "out-of-vocabulary" (OOV) en tweets escritos en español. Para corregir OOV incorrectos, el sistema de normalización genera candidatos "in-vocabulary" (IV) que aparecen en diferentes recursos léxicos y selecciona el más adecuado. Nuestro método genera dos tipos de candidatos, primarios y secundarios, que serán ordenados de diferentes maneras en el proceso de selección del mejor candidato.
Palabras clave: Normalización léxica, Mensajes cortos de texto, Procesamiento de tweets

Abstract: This paper describes a strategy to perform lexical normalisation of out-of-vocabulary (OOV) words in Spanish tweets. To correct any ill-formed OOV, the normalisation system generates in-vocabulary (IV) candidates found in several lexical resources, and selects the best one. Our method generates two types of candidates, primary and secondary IV candidates, which will be ranked in different ways to select the best candidate.
Keywords: Lexical Normalisation, Short Text Message, Tweet Processing

1 Introduction

In this paper, we describe a strategy to perform lexical normalisation of out-of-vocabulary (OOV) words in Spanish tweets. The task can be described as follows. Given an OOV, the algorithm must decide whether the OOV is either correct or ill-formed and, for the latter case, it must propose an in-vocabulary (IV) word found in a lexical resource to restore the incorrect OOV.

There has been few work on lexical normalisation in short messages. So far, the most successful strategy to normalise English tweets is described in (Han y Baldwing, 2012b; Han y Baldwing, 2013). They propose merging two different strategies: normalisation dictionary lookup and selection of the best in-vocabulary (IV) candidate.

The first strategy simply consists in looking up a normalisation dictionary, which contains specific abreviations and other types of lexical variants found in the Twitter language. Each lexical variant is associated to its standard form, for instance $gl \rightarrow girlfriend$. The dictionary lookup method achieves very high precision, but with low recall. As recall relies on the size of the dictionary, (Han y Baldwing, 2012a) propose to build wide-coverage normalisation dictionaries in an automatic way, by considering that lexical variants occur in similar contexts to their standard forms. Normalisation dictionary should only contain unambiguous "variant-standard" pairs. Ambiguous variants will be tackled using the second strategy.

The second strategy is applied when the OOV is a lexical variant that has not been found in the normalisation dictionary. It consists of the following two tasks:

- Generation of IV candidates (standard forms) for each particular OOV (lexical variant).
- Candidate selection of the best IV candidate.

The objective of the first task is to build, for each OOV, a list of standard forms which were derived from the OOV using different processes. For instance: reduction of character repetitions (e.g., $carrrr \rightarrow car$), or generation of those IV words whose Edit distance

* This work has been supported by Ministerio de Ciencia e Innovación, within the project OntoPedia, ref: FFI2010-14986.

with regard to the target OOV is within a given threshold.

The second task consists in selecting the best candidate out of the list generated in the previous step. Two different selection methods can be used: string similarity and context inference. To compute string similarity between the OOV and the different IV candidates, several measures and strategies can be used: lexical Edit distance, phonemic Edit distance, the longest common subsequence, affix substring, and so on. For context inference, the IV candidates of a given OOV can be ranked and then filtered on the basis of their local contexts. Local contexts are compared against a language model. The main problem of this method is that the local context of an OOV is often constituted by other incorrect lexical variants that are not found in the language model.

These two selection methods (string similarity and context inference) are complementary and then can be used together to select the best candidate.

There are, at least, two significant differences between the task evaluated in (Han y Baldwing, 2013) and that proposed at the Tweet Normalization Workshop at SEPLN 2013. On the one hand, the task in (Han y Baldwing, 2013) relies on the basic assumption that lexical variants have already been identified. This means that only ill-formed OOV are taken as input of the selection process. By contrast, the task defined by the Workshop guidelines includes the detection of ill-formed OOV. On the other hand, in (Han y Baldwing, 2013) the correspondences one-to-several are not considered, for instance *imo* → *in_my_opinion*. At the Workshop, by contrast, it is required to search for one-to-several correspondences, since the IV standard forms used to correct OOV can be multiwords. In sum, the task defined at the Tweet Normalization Workshop is more complex than that described in (Han y Baldwing, 2013).

Finally, there are other approaches to SMS and tweet normalisation based on very different strategies. For instance (Beaufort et al., 2010) and (Kaufmann y Kalita, 2010) make use of the Statistical Machine Translation framework, as well as of the noisy channel model, very common in speech processing. The main problem of these approaches is that they rely on large quantities of labelled training data, which are not available for microblogs.

2 The method

The normalisation method we propose combines the main strategies and tasks described in (Han y Baldwing, 2013), namely: normalisation dictionary lookup, generation of IV candidates, and selection of the best IV candidate with context information. In addition, given the conditions of the Workshop, we also include in our algorithm ill-formed OOV detection.

The design of our algorithm was motivated by the conclusions we draw from the analysis of the development corpus. We observed that the most frequent types of incorrect Spanish OOV are the following: (1) Uppercase/lowercase confusion: *patri* → *Patri* ; (2) character repetition for emphasis: *Buuenoo* → *Bueno* ; (3) language-dependent spelling problems, namely for Spanish: missing accents and letter confusion (v/b, g/j, ll/y, h/$\emptyset$ ...).

These three types of errors can be solved using simple specific rules. For the remaining phenomena, which correspond to more heterogeneous problems, we will make use of generic strategies such as those described in the previous section: dictionary lookup and selection of the best IV candidate. For detection of correct/incorrect OOV, we use the following method: if no IV associated to an OOV is found using specific rules or generic strategies, then the OOV is considered as correct. Otherwise it is taken as an ill-formed OOV. Text is lemmatised and PoS tagged using FreeLing (Padró y Stanilovsky, 2012).

Our method contains two modules: a set of lexical resources and an algorithm to detect and correct ill-formed OOV.

2.1 Lexical resources

Our system makes use of three different lexical resources:

ND Normalisation dictionary, containing incorrect lexical variants and their standard forms.

SD Standard dictionary, a list of correct forms generated from the lemmas found in the Real Academia Española dictionary (DRAE).

PND Proper names dictionary, containing

proper names extracted from the Spanish Wikipedia.

In the following, we describe how these three dictionaries have been built.

2.1.1 Normalisation Dictionary (ND)

It was mainly built using the developement data distributed by the organizers for the Tweet Normalization Workshop at SEPLN 2013. We also used as source of data the list of emoticons accesible from `http://en.wikipedia.org/wiki/List_of_emoticons`, as well as the list of Spanish abreviations released in `http://www.rae.es/dpd/apendices/apendice2.html`. Our final normalisation dictionary contains 824 entries.

2.1.2 Standard Dictionary (SD)

The standard dictionary is constituted by all the forms automatically generated from the lemmas found in DRAE. These lemmas have been extracted and freely distributed by the project `http://olea.org/proyectos/lemarios`. Verb forms were generated with the Cilenis verb conjugator (Gamallo et al., 2013), whereas we used specific morphological rules to generate noun and adjective forms. The final dictionary consists of 778,149 forms, which is significatively larger than that provided by the last version of FreeLing (556,509 Spanish forms in FreeLing 3.0).

2.1.3 Proper Names Dictionary (PND)

To make easier the detection of correct OOV (for instance, proper names and domain-specific terms that are not in a standard vocabulary), it is useful to make use of a large list of OOV extracted from an enclyclopaedic resource, for instance the Wikipedia. Several PND were automatically extracted. Finally, the PND allowing the best performance in the normalisation task was extraced as follows: First, using CorpusPedia (Gamallo y González, 2010), a simplified format derived from the original downloadable XML file (Wikipedia Dump of May 2011), the names of articles belonging to categories related to persons, locations, and organisations were identified, by using the strategy described in (Gamallo y Garcia, 2011). Then, these names were tokenized and those unigrams whose lowercase variants are found in the standard dictionary (SD) were filtered out. The result is a list of 107,980 unigrams taking part in the names of persons, locations, and organisations.

2.2 The algorithm

The system takes a list of OOV as input. An OOV is considered as correct if the *Dictionary Lookup* process is true. Dictionary lookup is a process that consists in searching a token in one of the three lexical dictionaries: ND, SD, or PD. If the OOV is found in one of them, then it is considered as correct. However, even if Dictionary Lookup is false, the OOV will be considered as correct if *Affix Check* is true. Affix Check is a process that extracts regular suffixes and prefixes from the OOV and verifies whether the stem of the OOV takes part of an entry found in one of the three dictionaries. Otherwise, the OOV can be incorrect.

Given an incorrect OOV, we generate a list of variants. A variant of an OOV is an IV candidate if either Dictionary Lookup or Affix Check is true. We distinguish between primary and secondary variants.

2.2.1 Generation of primary variants

Primary variants of an OOV are its most likely IV candidates, according to the type of errors we found in the development corpus. Primary variants will be favoured in the process of candidate selection: if at least a primary variant is found, then the system does not consider secondary variants.

Primary variants of an OOV are those IV candidates derived from the OOV that only differ from the source OOV with regard to one of these linguistic phenomena: Uppercase/lowercase confusion, character repetition, or frequent Spanish spelling errors. The frequent spelling errors include, not only typical problems with accents and frequent letter confusions (v/b, j/g, etc), but also some phonemic conventions, namely the use of "x" for "ch" (e.g. *xicle* → *chicle*). Primary variants generated by simplifying repetition include the cases of interjection reduction: *jejeeje* → *je*. For uppercase and lowercase variation, we take into account that words can be written with only lowercase letters, with capitalisation (proper names or first position in the sentence), or with only uppercase letters (e.g. acronyms). For instance, given the OOV "pedro", two other variants are generated: "Pedro" and "PEDRO". If one of them is found in the lexical resources,

then it is considered as a primary IV candidate. Let us note that a primary variant is considered an IV candidate if either Dictionary Lookup or Affix Check is true.

2.2.2 Generation of secondary variants

If no primary variant is found as IV candidate, then a large list of secondary variants is generated using Edit distance. In our experiments, we only generate those variants that have Edit distance 1 with regard to the original OOV. Dictionary Lookup and Affix Check allow us to identify the list of secondary IV candidates. In the next step, we select the best candidate.

2.2.3 Candidate selection

To select the best IV candidate of a given OOV, we compare the local context of each candidate against a language model containing bigrams of tokens found within a window of size 4 (2 tokens to the left and 2 to the right of a given token). More precisely, for each candidate, chi-square measure is computed by considering observed frequencies in the local context against expected frequencies in the language model. The language model was built by selecting lemmas of the following list of PoS categories: nouns, verbs, adjectives, prepositions, and adverbs. Text was processed with FreeLing. We also introduced an important restriction that takes into account whether the IV candidate is either a primary or a secondary variant. A primary variant is always selected even if its chi-square score is 0. It means that a primary variant is always selected even if it is not found in the language model. By contrast, for secondary variants, the chi-square must be higher than 0 to be selected. Candidates are ranked considering chi-square values and the above restriction. The best IV candidate on the top of the rank is selected and given as correction of the OOV. At the end, we apply the capitalisation rule which considers the position of the original ill-formed OOV in the sentence: if it is the first word in the sentence, then the selected IV candidate must be written with its first letter in uppercase.

Finally, if no IV candidate (primary or secondary variant) is selected, then the OOV is considered as correct. So, correct OOV are detected in two different ways: first, if Dictionary Lookup or Affix Check is true for the original OOV, or if no IV candidate is selected.

3 Experiments

Some experiments were performed using as test set the development corpus provided by the organisation of the Tweet Normalization Workshop. This corpus contains 500 tweets and 651 OOV manually corrected. The language model used by our system was built from two text sources: the collection of 227,255 tweets provided by the Workshop, which were captured between April 1st and 2nd of 2013, and a collection of news from El Pais and El Mundo captured via RSS Crawling. In sum, the language model was created from 50MB of text. The normalisation dictionary contains annotated information from the sample corpus with 100 tweets provided by the Workshop. For the final tests, this dictionary also includes the annotated pairs of the development corpus.

Two versions of our system were tested, "Standard" and "Restricted", and compared against two baselines: "Baseline1" and "Baseline2". The standard version has been described in the previous section. The restricted version includes a constraint on short proper names and short acronyms (with less than 5 letters). The constraint prevents short proper names and acronyms from being expanded with secondary variants. For instance, if the OOV is "BBC", the system does not create IV candidates such as "BBV", "ABC", and so on. In Baseline1, we do not separate primary from secondary variants, and all IV candidates are treated as primary variants. Baseline2 does not separate primary from secondary variants, and all IV candidates are treated as secondary variants.

Table 1 shows the results obtained from the experiments performed on the development set. The best performance is achieved with "Restricted", which is based on the algorithm that makes use of restrictions on short proper names. The low scores reached by the baseline systems clearly show that candidates must be separated at different levels to be treated in different ways. In the test set, "Restricted" achieved 66.3% accuracy, the second best score among the 13 participants in the Tweet-Norm Competition.

Bibliografía

Beaufort, Richard, Sophie Roekhaut, Louise-Amélie Cougnon, y Cédrick Fairon. 2010.

Systems	pos	neg	accuracy
Baseline1	273	378	41.80
Baseline2	288	363	44.10
Standard	444	207	67.99
Restrictive	451	200	**69,06**

Table 1: Results from the development set

A hybrid rule/model-based finite-state framework for normalizing SMS messages. En *48th Annual Meeting of the Association for Computational Linguistics*, páginas 770–779, Uppsala, Sweden.

Gamallo, P., M. Garcia, I. González, M. Mu noz, y I. del Río. 2013. Learning verb inflection using Cilenis conjugators. *Eurocall Review*, 21(1):12–19.

Gamallo, Pablo y Marcos Garcia. 2011. A resource-based method for named entity extraction and classification. *LNCS*, 7026:610–623.

Gamallo, Pablo y Isaac González. 2010. Wikipedia as a multilingual source of comparable corpora. En *LREC 2010 Workshop on Building and Using Comparable Corpora*, páginas 19–26, Valeta, Malta.

Han, B. y T. Baldwing. 2012a. Automatically constructing a normalisation dictionary for microblogs. En *Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL 2012)*, Jeju, Korea.

Han, B. y T. Baldwing. 2012b. Lexical normalisation of short text messages: Makn sens a twitter. En *49th Annual Meeting of the Association for Computational Linguistics*, páginas 368–378, Portland, Oregon, USA.

Han, B. y T. Baldwing. 2013. Lexical normalisation of social media text. *ACM Transactions on Intelligent Systems and Technology*, 4(1):15–27.

Kaufmann, J. y J. Kalita. 2010. Syntactic normalization of twitter messages. En *Conference on Natural Language Processin*, Kharagpur, India.

Padró, Lluís. y Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. En *Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.