The TALP-UPC approach to Tweet-Norm 2013

La participación de TALP-UPC en Tweet-Norm 2013

Alicia Ageno y **Pere R. Comas** y **Lluís Padró** y **Jordi Turmo**
TALP Research Center
Universitat Politècnica de Catalunya — UPC
{ageno,pcomas,padro,turmo}@lsi.upc.edu

Resumen: Este artículo describe la metodología utilizada por el equipo TALP-UPC en la tarea propuesta en SEPLN 2013 para la normalización de tweets (Tweet-Norm). El sistema usa una batería de módulos para generar diferentes propuestas de corrección para cada palabra desconocida. La corrección definitiva se elige por votación ponderada según la precisión de cada módulo.
Palabras clave: Normalización de texto, Corrección de texto

Abstract: This paper describes the methodology used by the TALP-UPC team for the SEPLN 2013 shared task of tweet normalization (Tweet-Norm). The system uses a set of modules that propose different corrections for each out-of-vocabulary word. The final correction is chosen by weighted voting according to each module accuracy.
Keywords: Text normalization, Text correction

1 Introduction

The increasing use of social networks to briefly express opinions and facts is leading to large amounts of text written with misspellings and neologisms, such as the case of tweets. The SEPLN 2013 Tweet-Norm shared task focuses on the evaluation of approaches useful for normalizing out-of-vocabulary words occurring in Spanish tweets, similar to the previous works such as (Han and Baldwin, 2011) for English and (Mosquera, Lloret, and Moreda, 2012) for English and Spanish. In this paper we describe the UPC system for this task and the results achieved.

2 Our approach

The UPC system for SEPLN 2013 Tweet-Norm shared task consists of a collection of *expert* modules, each of which proposes corrections for out-of-vocabulary (OOV) words. The final decision is taken by weighted voting according to each expert accuracy on the development corpus.

First, a preprocessing step is applied, where consecutive occurrences of the same letter are reduced to one (except valid Spanish digraphs like `rr` or `ll`). We generate also a version of the OOV with those repetitions reduced to two occurrences (to capture cases such as `coordinar`, `leed`, `acción`, etc.). In this way, we obtain three different OOV versions (original, reduction to one repeated letter, reduction to two repeated letters) that will be checked against dictionaries and gazetteers as described below.

All expert modules are implemented using FreeLing (Padró and Stanilovsky, 2012) library facilities for dictionary access, multiword detection, or PoS tagging. Some experts use string edit distance (SED) measures to find words in a dictionary similar to the target OOV. FOMA library (Hulden, 2009) is used in these cases for fast retrieval of candidates.

The used expert modules can be divided in three classes:

- **Regular-expression experts**: Experts in this class are regular expression collections that propose corrections for recurring patterns or words, such as smileys, laughs (e.g. `jajjaja`, `jeje`, etc.), frequent abbreviations (e.g. `TQM` → `te_quiero_mucho`, `xq` → `porque`, etc), or frequent mistakes (e.g. `nose` → `no_sé`). Experts in this category propose a fixed solution for each case.
- **Single-word experts**: Each module belonging to this class uses a specific single-word lexical resource and a set of string edit distance (SED) measures to find candidates similar to the target word. The three SED measures specif-

ically used for the task are: character distance (the conventional edit distance metric between strings), phonetic distance (transformations according to similarity in pronunciation) and keyboard distance (transformations due to possible errors when typing).

- **Multi-word experts**: Modules in this category take into account the context where an OOV is located to select the best candidate among those proposed by the other experts. We used three different experts in this category. First, the *multiword dictionary* module takes into account proposals of the single-word experts that use different distances over a dictionary consisting only of tokens that appear in known multiwords. All combinations of possible candidates for the OOV and its context are checked against the multiwords dictionary, and those matching an entry are suggested as corrections. Second, the *PoS tagger* expert takes into account all proposals of all single-word experts, retrieves the possible PoS tags for each of them, and creates a virtual token with a morphological ambiguity class including all obtained categories. Then, a PoS tagger is applied, and the best category for each OOV is selected. The module filters out all proposals not matching the resulting tag, and produces as candidates only those with the selected category. Finally, the *glued words* expert, which consists of a FSM that recognizes the language $L(\_L)+$, where L is the language of all valid words in the Spanish dictionary. Using foma-based SED search on this FSM with an appropriate cost matrix, we can obtain, for instance, that `lo_siento` is the word in the FSM language closer to `losiento`, and propose it as a candidate correction.

All the resources used in these experts are briefly enumerated in Section 3.

After all experts have been applied, a selection function is used on the set of resulting candidates. This selection function takes into account the SED distance of each proposal to the original OOV, the number of experts that proposed it, and the precision, recall, and F_1 of each expert on the development corpus to perform a weighted voting and select the final correction. Different experiments with different functions are reported in Section 4.

3 The set of lexical resources

In this section, we describe the different lexical resources we have employed in order to provide correct candidates for OOV words. Some of these resources are merely looked up with exact search, whilst for others we have considered useful to perform approximate search as well, using the SED metrics described in the previous section. Note, in addition, that the unnanotated tweets provided by the organization have not been used to enrich our resources.

Next, we list each of these resources as well as the types of searches for which they are used.

3.1 Resources for regular-expression experts

- **Gazetteer of acronyms**: List of different sorts of acronyms. It also includes several abbreviations frequently used in tweets and other short messages. Used only for exact searches.
- **Gazetteer of emoticons**: List of emoticons, some of them expressed as regular expressions. We just deal with those emoticons which are not composed by only punctuation signs, since the other ones would be accepted by FreeLing, and consequently they will not be OOV words.
- **Gazeteer of onomatopoeias**: List of onomatopoeias, many of them expressed as regular expressions. The RAE dictionary is used as the reference for the correct normalization of each candidate onomatopoeia found.

3.2 Resources for single-word experts

- **Spanish dictionary**: List of Spanish words, according to FreeLing dictionary. The three types of SED metrics are performed on it.
- **English dictionary**: List of English words, according to FreeLing dictionary. The three types of SED metrics are also performed on it.
- **Spanish dictionary expanded with morphological derivates**: A set of

morphological derivates has been generated for the words in the Spanish dictionary. The specific derivates have been applied according to the PoS of each word. Concretely, superlatives and diminutives have been generated for nouns, adjectives, adverbs and participles, and enclitic pronouns have been suffixed to infinitive and imperative verbal forms, as well as to gerunds. However, due to the high volume of generated alternatives, a previous filter based on commonness and length of the words has been performed on the dictionary and only the derivates for the resultant words have been generated. On this resource, the exact search and the three types of SED metrics are performed.

- **Gazetteer of names**: List of person names (including also certain diminutives). Both the exact and the SED metrics are carried out on it.
- **Uniwords NE gazetteer**: It comprises a far from exhaustive list of proper nouns such as different types of locations, companies, artists and other personalities, TV channels and programs, products, newspapers and media groups or even shopping centers. As mentioned in the previous section, this gazetteer is used both as a preliminary search for the elements of the multiwords gazetter and as a gazetter in itself. In the latter case, exact search and SED metrics for approximate searches are performed on it.

3.3 Resources for multi-word experts

- **Multiwords NE gazetteer**: It comprises a far from exhaustive list of proper nouns composed by more than one word, belonging to the same categories mentioned for the uniwords gazetteer. Neither exact search nor SED metrics are performed directly on it. As mentioned in the previous section, they are performed in the uniwords gazetteer.

4 Experiments on different functions for the best candidate

Combining the experts from Section 2 and the lexical resources described in Section 3, we obtain a total of 32 different producers that are integrated in our tweet normalizer. Additionally, we add a 33rd producer that always proposes to leave the target OOV as it is. The combined outputs of these producers yield several hundreds of spelling alternatives for the OOV words, therefore we need a principled method to choose the best one among them, including to leave the original word as it is given. This strategy is able to propose the correct spelling alternative to 89.42% of the OOVs found in the development corpus, therefore, this is the upper-bound accuracy of our system.

Using the development corpus, we have computed the precision, recall and F_1 of each producer. Since the producers yield a list of spelling alternatives that are sortable according to the SED metrics, we have devised three different levels where we can measure its confidence:

- **TopN**: At this level, we check only if the producer produces the correct correction anywhere in the alternatives list.
- **Top1**: This level checks how many times the correct correction has the smallest SED in the whole list of alternatives (i.e., it is on the front of the list). In this case, the precision is computed against the total number of proposed corrections having the smallest SED.
- **Top0**: This measures how many times the correct correction has a SED distance of zero over the total number of proposals at distance zero. Note that all exact searches (regular-expression experts and look up dictionaries) yield alternatives with distance zero.

We compute precision, recall and F_1 for each producer for all three levels of measure.

To produce a proposal for each OOV, we implement a voting scheme. Each producer votes for each of their proposed corrections using the suitable TopN, Top1 or Top0 scores as their vote weight. The possible corrections are pooled together and the one with the largest total score is our final proposal.

Note that a proposed correction in the Top0 position is also in the Top1 and the TopN positions. Therefore, we can choose if the weight of a producer vote is just the score of it's best measure (e.g. Top1 instead of TopN) or the addition of all suitable measures (e.g. Top1 plus TopN for a proposal in Top1). We have experimented with these two

weight	scheme	normal	squared
R	single	54.79	65.92
R	additive	60.77	69.12
P	single	65.78	67.45
P	additive	67.45	69.81
F_1	single	65.09	67.87
F_1	additive	67.87	69.12
w100	single	59.52	—
w110	single	41.16	—
w111	single	23.78	—
w111	additive	45.61	—

Table 1: Choosing the best voting scheme

voting schemes that we call *single* or *additive* and with using precision (P), recall (R) or F_1 as the actual vote weight. We have also considered the possibility of squaring the weights in order to strengthen the relevance of high precision producers. Table 4 shows the results achieved using the development corpus for testing and estimating the weights. We have set up some baselines giving fixed weight to the votes. We have the scheme *w111*, which gives a weight of 1 to TopN, Top1 and Top0; the scheme *w110* gives a weight of 0 to TopN and of 1 to the other two, and so on.

The experiments show that using squared precision as the confidence measure within an additive scheme yields the best results: a precision of 69.81% on the development corpus.

5 Results

The official result of our single run is an accuracy of 65.26% on a test corpus of 500 tweets containing roughly 700 OOVs. In this run we use the weights estimated from the development corpus. This results is 4.5 points behind what we obtained on the development corpus, suggesting that our estimation method is reasonable but may be overfitting.

To elucidate this issue, we have repeated our experiment using the test gold standard to estimate the vote weights (instead of using the development data). With this setup and identical voting scheme, the precision is increased by 0.76 points, less than four points behind the 69.81% we got in the development set. Additionally, we have used the gold standard to calculate the upper-bound of our producers as we did with the development data. We are able to propose the correct word for 85.47% of the OOVs, which is 4 poins behind the 89.42% for development data.

Since little improvement is obtained when using the test data, this suggests that our strategy of estimating each producers' precision is not overfitting. Additionally, we can see how the drop in the system's upper-bound matches its accuracy drop. Therefore, we believe that the nature and distribution of OOVs in Twitter streams may vary over time more than it is represented on the the development set, thus, our strategy as a whole is more suited to this particular set of development data than to the test data.

6 Conclusions and future work

In this paper, we have described our approach for the SEPLN 2013 Tweet-Norm shared task, which consists in normalizing a set of predefined out-of-vocabulary words occurring in tweets. Our system is based on a voting schema that combines 33 different experts on OOV candidate selection, each one using a specific viewpoint defined by a particular pair of edit distance similarity metric and lexical resource. This approach achieved a precision of 65.26% in the test corpus, ranking our system in the 3rd best place among the participants. This result show the appropriateness of our approach for the task. However, it is far to achieve the upper-bound results (i.e., from the 69.81% achieved for the development corpus to the upper-bound of 85.47% achievable in that corpus). This fact shows that there is room enough to improve our system.

In order to get improvement, main lines in our future work involve enriching the lexical resources with OOV words occurring in the unnanotated tweets provided by the organizers, using a richer context of the OOV words to drop out false candidates, tuning the costs of the edit distances operators, and considering other alternative voting schemes.

Acknowledgments

This research has been partially funded by the Spanish research project SKATER (TIN-2012-38584-C06-01) and the EU FP7 Programme project XLike (FP7-ICT-2011.4.2).

References

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 Associa-*

tion for Computational Linguistics (ACL 2011).

Hulden, Mans. 2009. Fast approximate string matching with finite automata. *Procesamiento del Lenguaje Natural*, (43):57–64.

Mosquera, Alejandro, Elena Lloret, and Paloma Moreda. 2012. Towards facilitating the accessibility of web 2.0 texts through text normalisation. In *Proceedings of the LREC 2012 Workshop: Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*.

Padró, Lluís and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*. ELRA.