Noname manuscript No.
(will be inserted by the editor)

TweetNorm: A Benchmark for Lexical Normalization of Spanish Tweets

Iñaki Alegria[1] · Nora Aranberri[1] · Pere R. Comas[2] · Víctor Fresno[3] · Pablo Gamallo[4] · Lluis Padró[2] · Iñaki San Vicente · Jordi Turmo[2] · Arkaitz Zubiaga[6]

Received: date / Accepted: date

Abstract The language used in social media is often characterized by the abundance of informal and non-standard writing. The normalization of this non-standard language can be crucial to facilitate the subsequent textual processing and to consequently help boost the performance of natural language processing tools applied to social media text. In this paper we present a benchmark for lexical normalization of social media posts, specifically for tweets in Spanish language. We describe the tweet normalization challenge we organized recently, analyze the performance achieved by the different systems submitted to the challenge, and delve into the characteristics of systems to identify the features that were useful. The organization of this challenge has led to the production of a benchmark for lexical normalization of social media, including an evaluation framework, as well as an annotated corpus of Spanish tweets –TweetNorm_es–, which we make publicly available. The creation of this benchmark and the evaluation has brought to light the types of words that submitted systems did best with, and posits the main shortcomings to be addressed in future work.

Keywords Lexical normalization, Twitter, social media, corpus, evaluation

1 Introduction

With its evergrowing usage as a microblogging service, Twitter has become a ubiquitous platform where users continuously share information in a real-time fashion. Information is posted by users in the form of *tweets*, which are characterized by their brevity, restricted by Twitter's 140 character limit, and which often lack correct grammar and/or spelling. This posits the need for a process of lexical normalization of these tweets as a key initial step for subsequently applying natural language

(1) IXA. UPV/EHU, (2) UPC, (3) UNED, (4) USC, (5) Elhuyar, (6) University of Warwick
tweet-norm@elhuyar.com

processing (NLP) tools such as information extraction, machine translation and sentiment analysis. Even though research on lexical normalization of tweets is still in its infancy, early studies have shown that it can indeed boost performance of NLP tools that work on tweets [34]. While lexical normalization of SMS and tweets in English has attracted the interest of a community of researchers [10,11,21], little has been studied for this kind of short texts written in other languages such as Spanish.

A lexical normalization system takes a natural language sentence as input, and consists of the following two stages: (i) non-standard word detection, which identifies the words from the input sentence that need to be normalized, and (ii) candidate selection, which selects the alternative word as the normalized output. As a result, the objective of a lexical normalization system is to output a modified version of the input sentence, such that non-standard words have been normalized. Both stages are crucial to build an accurate normalization system, since a wrong decision in the first step as to whether a word needs to be normalized, will lead to a bad candidate selection in the subsequent step and thus to an inaccurate normalization of the word. This inaccurate normalization can be twofold. On one hand, mislabeling a word as "non-standard" will lead to the wrong detection of a candidate and, on the other hand, mistakenly identifying a non-standard word as being correct will skip the subsequent candidate selection stage when it is really needed.

In order to motivate additional research in the field, we organized the Tweet-Norm 2013 shared task[1] held at the 29th Conference of the Spanish Association for Natural Language Processing[2] (SEPLN). The goal of the shared task was to create a benchmark for lexical normalization of microtexts written in Spanish, including both a robust evaluation framework, and an annotated corpus to perform the experiments with. In this shared task, participants were provided with such corpus and evaluation guidelines, and were asked to normalize a set of tweets containing several non-standard word forms each.[3]

We created a corpus of tweets annotated with normalized variations, which we released to participants of the shared task for benchmark evaluation purposes. The creation and distribution of a benchmark corpus provided a common testing ground, which enabled us to compare performance of participants and to identify the main advantages and shortcomings of each participating system. This paper makes the aforementioned corpus publicly available with the aim of attracting researchers and practitioners to develop new normalization approaches while making use of a common evaluation setting. We describe the methodology followed for the collection of tweets and the generation of the annotated corpus that has been put together in the resulting *TweetNorm_es* corpus. We also present a detailed analysis of the results of the shared task, delving into the performance of each system and breaking down performance values into word categories, including common words, onomatopoeias, entities, and others. This detailed analysis allows us to shed some light on the types

[1] Details about the workshop can be found at `http://komunitatea.elhuyar.org/tweet-norm/`

[2] `http://nil.fdi.ucm.es/sepln2013/`

[3] The term "ill-formed" has also been used in the literature to refer to these non-standard word forms. We opted for the term "non-standard word form" because some of the words that fall into this category, such as abbreviations or acronyms, are not necessarily misspellings.

of approaches that can be of help to build accurate normalization systems, and to set forth the main shortcomings that need to be addressed in future research in the field.

The corpus can be used, modified and redistributed under the terms of the CC-BY license.[4]

2 Related Work

Twitter is being used increasingly as an information source for NLP research in multiple tasks. These include sentiment classification [9,16], topic modeling [19,13], and summarization [15,4], among many others. However, Twitter poses an unprecedented problem for NLP research; the fact that users tend to shorten their texts to make it fit into a tweet, often with the extra challenge of posting from a mobile device, and the occasional misspellings and typos they introduce, leads to the creation of short texts characterized by a non-standard language, which makes the analysis of texts more challenging. Eisenstein [7] outlines the challenges posed by the *bad language* that characterizes the Internet, and surveys two of the most popular directions in which the NLP community has tackled this issue: normalization and domain adaptation.

However, the lexical normalization of tweets is still in its infancy as a research field. Some of the early work in the field by Han et al. [10,11], comparing different approaches for lexical normalization, found that approaches based on language models perform significantly worse than dictionary lookup methods. This is likely due to the fact that the lexical context in Twitter data is noisy; on many occasions, Out-of-vocabulary (OOV) words can co-occur with user mentions, hashtags, URLs, and even other OOV words, which can produce poor context information. They also observed that well-known methods for normalization in other domains suffer from the poor performance of lexical variant detection, which worsens the effectiveness of existing techniques to the context of Twitter and social media. The best system in Han and Baldwin [10], which is mainly based on dictionary lookup, achieved an F-score of 75.3% in a partial evaluation that focused only on the candidate selection step, assuming that the previous step for non-standard word detection was perfect.

Motivated by the performance of the dictionary-based normalization system, in a later study Han et al. [11] enhanced their system by using information from both word distribution and string similarity to build normalization lexica with broader coverage. They reported an F-score of 72.3% when dealing with the whole task, i.e., including both the OOV detection and normalization steps. These results can be considered to be the state-of-the-art for normalization of English tweets. Their case study, albeit focused on English tweets, is straightforwardly applicable to other languages, given that they defined a generalizable research methodology for this kind of tasks. In the present work, we relied on their methodology to define the corpus annotation guidelines, as well as to set up the shared task. However, there are a few differences that we introduced in our case:

- As we said above, most previous work assumed perfect OOV detection, and focused on the subsequent candidate selection step. Instead, the shared task held at

[4] `http://creativecommons.org/licenses/by/3.0/legalcode`

Tweet-Norm 2013 considers both the detection of OOVs, and the normalization as a single process. To our knowledge, only Han et al. [11] have proposed such an integral solution.
- Different from both [10, 11], the shared task at Tweet-Norm 2013 also considers multiwords in the lexical normalization process. We considered one-to-many correspondences (e.g., $imo \rightarrow in\_my\_opinion$), and so the submitted systems had to deal with multiwords.

Using several corpora, including the one described above, Liu et al. [21] propose a normalization system, and especially focus on exploring the coverage achieved by this system when applied to SMS and Twitter data. They propose a cognitively-driven normalization system that integrates different human perspectives into the normalization of non-standard tokens, including enhanced letter transformation, visual priming, and string/phonetic similarity. Results show that the presented system achieves over 90% word-coverage across all datasets.

Others have also performed a preliminary tweet normalization step prior to the main task. For instance, Wei et al. [34] perform a 4-step normalization of English tweets before running their topic detection system: (i) OOV word detection, (ii) slang word translation, (iii) candidate set generation, and (iv) candidate selection. They performed an *in vivo* evaluation of their system, looking instead at the performance boost of the system presented to the TREC 2011 microblog search track. They found the normalization system to be effective, providing a slight improvement to the results, although a more comprehensive normalization system could do even better. Similarly, Liu et al. [22] used a tweet normalization system as the initial step of their system for named entity recognition. They use statistical learning algorithms, trained with the pairs provided by Han and Baldwin [10], as well as 1,500 more pairs which were compiled manually. They obtained an F-score of 60.5%.

Others have opted for making use of large-scale data collections to train their normalization systems. Examples include Beaufort et al. [3], who tackle the task of normalizing SMS texts using the noisy channel model, very common in speech processing, or Kaufmann and Kalita [17], who feed a statistical machine translation model with tweets, to turn them into standard English. Another example is the work by Ling et al. [20], who make use of self-translation from Twitter and Sina Weibo in order to obtain large-scale (albeit noisy) normalization examples. The Mandarin version is automatically translated back into English, and then two versions are available in English: the original (not normalized) and the noisy translation from the equivalent tweet in Mandarin (noisy normalized). Then, they use the SMT framework for learning the normalization patterns. However, despite research in this direction, in this work we are interested in avoiding the need for large-scale training data, as this tends to be costly. Moreover, no such resources are available to the best of our knowledge for microblogs, in particular for languages other than English. Hence, we wanted to define the tweet normalization task assuming the limited availability of training data, and thus allowing participants of the task to focus on the algorithms and external resources that can be of help.

Wang et al. [33] tackled a related task for microtext normalization task, in this case in Chinese. Instead of normalizing the spelling of words, they studied the translation of Chinese texts into their formal alternatives. This is especially important given that machine translation systems tend to mistranslate informal words when translating for instance into English. The authors studied first the linguistic phenomenon of informal words in the domain of Chinese microtext, and presented then a method for normalizing Chinese informal words to their formal equivalents. The task is formalized as a classification problem and proposes rule-based and statistical features to model three channels or phenomena (i.e., phonetic substitution, abbreviation and paraphrase) that identify connections between formal and informal pairs. They created a corpus for evaluation purposes, which was annotated through crowdsourcing, and reported a precision score of 89.5%.

To the best of our knowledge, previous efforts have focused on language specific approaches for English tweets, there is limited work for Chinese in a related task, and no work has dealt with tweets written in Spanish. Our work intends to fill this gap by tackling the normalization for Spanish tweets, defining a benchmark for evaluation. To the best of our knowledge, this is the first work that deals with lexical normalization of non-English tweets. As a related effort for tweets in Spanish, Villena et al. [32] organized a shared task focused on sentiment analysis. Costa et al. [5] and Oliva et al. [26] have also worked on normalization of SMS texts in Spanish.

3 Corpus

In this section, we describe the process we followed to collect and sample the tweets, which were manually annotated. First, we describe the API settings defined to collect the tweets, and then explain the preprocessing step carried out to prepare the data for manual annotation.

3.1 Tweet Dataset

Among the Twitter APIs[5] for tracking and collecting tweets, we opted for geolocated tweets, whose metadata include the coordinates of the location each tweet was sent from. Twitter's API allows the user to select tweets sent from a pre-determined geographic area. Making use of this feature, we chose an area within the Iberian Peninsula, taking out regions where languages other than Spanish are also spoken. We found this approach to be highly effective when it comes to gathering large numbers of tweets in Spanish. Thus, the selected geographic area forms a rectangle with Guadalajara (coordinates: 41, -2) as the northeasternmost point and Cádiz (coordinates: 36.5, -6) as the southwesternmost point. The collection of tweets gathered on April 1-2, 2013 amounts to 227,855 tweets. From this large dataset, we created two random subsets of 600 tweets each, which were shared with participants, one as a training set, and the other as a test set for final evaluation purposes. The rest of the dataset was also shared with participants, with no manual annotations, which they could use for setting up their unsupervised normalization systems.

[5] https://dev.twitter.com/docs/api

3.2 Preprocessing

We used the FreeLing[6] language analysis tools [27] for the identification of OOV words from tweets. We used some of the basic processing modules included in this library to tokenize and analyze tweets. A token was ultimately considered to be an OOV when none of the modules identified it as an in-vocabulary (IV) word.

We used the following modules to process the tweets:

- tokenizer.
- usermap.
- punctuation detection.
- number detection.
- date detection.
- morphological dictionary (with affixes handling).
- quantitites detection.

These modules were set up with their default configuration, except in the following cases, for which we detail the changes we made:

- `tokenizer`: The rules of the tokenizer were tuned to keep usernames (`@user`), hashtags (`#hashtag`), e-mail addresses, URLs, and the most common emoticons as single tokens.
- `usermap`: We also enabled the `usermap` module (disabled in the default FreeLing configuration), which checks if each token matches one of a set of regular expressions that are discarded from being considered as OOVs. These regular expressions help identify usernames, hashtags, e-mail addresses, URLs, and common emoticons.

Specific configuration files used for `tokenizer` and `usermap` were later included in the FreeLing distribution and are available at the project's SVN repository.

On the other hand, the following modules for morphological analysis were disabled:

- multiword detector (to avoid agglutination of several tokens into a single one).
- named entity detector (since we want to keep them as OOVs).
- lexical probabilities module (which includes a guesser that would assign at least one analysis to each word).

4 Annotation Methodology

During the annotation process, experts were asked to annotate the OOV words. They tagged each OOV word either as *correct*, *variant* or *NoES* (not in Spanish). For those cases deemed variant, they also provided the normalized spelling of the word along with the annotation. Standard word forms are derived from the RAE dictionary[7].

[6] `http://nlp.cs.upc.edu/freeling`

[7] RAE, or *Real Academia Española*, is the institution responsible for regulating the Spanish language.

Three experts independently annotated each OOV word for the development set, and two of them participated in the annotation of the test corpus. We put together the annotations from the different experts by majority voting when possible, and by further discussing the correct annotation among the experts in case of ties. To facilitate the annotation process and subsequent discussions, we defined the following guidelines for each OOV word:

- When the word is included in RAE's dictionary: mark it as correct.
- When a well-formed word refers to a Named Entity (e.g., Zaragoza) or a loanword (e.g., Twitter): mark it as correct.
- When a word incorporates an emphatic or dialectal variation, it is misspelled, or lacks or misuses the acute accent: mark it as variation and provide the standard spelling (e.g., muuuuuuucho/mucho, kasa/casa, cafe/café).
- When more than one word is written together with no separation: mark it as variation and provide the standard spelling (e.g., asik/así_que, find/fin_de_semana).
- When a single word is split into smaller strings: mark all of them as variations and provide their standard spellings (e.g., im_presionante/impresionante, per_do_na_meeee/perdóname).
- When a word is unintelligible, a foreign word, or others (e.g., XD): NoES.

Note that the guidelines distinguish between "loanwords" and "foreign words". We consider "loanwords" those that, despite belonging to a language different from that of the tweet, have been assimilated by it and are used in everyday language (e.g. "tablet", "sandwich"). In contrast, "foreign words" are those that have not been assimilated, and therefore, do sound foreign (e.g. in the tweet "Igor gracias no sabia que te importara tanto joo tio!! pero es que eres mu feote sorry", "sorry" is a foreign word). Named entities, in turn, are treated separately.

These guidelines include the most common cases, but some of the cases we found were still not covered. In those cases, we met to further discuss each case in search of the most suitable solution.

Examples of uncommon cases not considered by the guidelines above include:

- `que estafa de tablet`
  [what a scam is this tablet]
  (in this case *tablet* is a loanword that is not included in the RAE dictionary yet, but the Spanish alternative *tableta* will incorporate this new meaning in the next release of the dictionary).
- `Me dispongo a ver Game of Thrones.`
  [I'm going to watch Game of Thrones]
  In this case the original name of the series was used instead of the Spanish translation *Juego de Tronos*).

One of the most challenging cases we identified during the annotation process was the normalization of abbreviations. In some cases, the context surrounding the abbreviated word in question is not sufficient to disambiguate its meaning and to identify the intention of the user. For instance:

- `cariiii k no te seguia en twitter!!!mu fuerte!!!..yasoy tu fan....muak....se te exa d menos en el` **`bk`**`....sobreto en los cierres jajajajas`
  [my dear i wasn't following you on twitter!!no way!!i'm so fan of you from now on....kisses...
  we miss you in the **bk**.... especially when closing hahaha]

where it is difficult to know what *bk* refers to with certainty. This addressee had seemingly a colleague at a place called *bk*, but there is little evidence to grasp its exact meaning without further research. The annotators ultimately chose *Burger King* as the variant, as the most likely choice for the acronym. In a few cases, OOVs could not be disambiguated and the annotators provided two alternatives. This includes cases where the gender could not be disambiguated from the abbreviated form –e.g., a tweet from the corpus contained the word *her*, which may refer to either *hermano* (brother) or *hermana* (sister).

The meaning of some onomatopoeias was also hard to grasp in some cases, which needed further discussion to come to an agreement among annotators. For instance:

- `me da igual JUUUM!!`
  [i don't care huuum!!]

5 Development and test corpora

Two collections have been generated from the initial corpus described in Section 3.1: the development corpus and the test corpus, which consist of 600 tweets each. A total of 775 and 724 OOV words were manually annotated respectively in both corpora.

As required by Twitter API's terms of use,[8] we do not release the content of the tweets, but provide instead the user names and tweet IDs that enable to download the content of the tweets by using *Twitid*. *Twitid*[9] is a script that retrieves the content of tweets from the list of user names and tweet IDs.

Since we distributed the lists of tweets to participants following the method above, chances are that some tweets might have become unavailable. Some tweets may become unavailable as users remove their accounts, make them private, or delete the tweet. This may lead to participants having slightly different collections of tweets, which would affect the evaluation process. We figured this out by identifying the subset of tweets that were still available after all participants submitted their results. We found that 562 of the 600 tweets in the original test set were still accessible at the time. Thus, the initial set of 724 OOV words found in the initial test corpus was reduced to 662 due to the unavailable tweets. We relied on this slightly reduced set of tweets for the final evaluation.

Both datasets are publicly available[10] under the terms of the CC-BY license. The datasets include tweet IDs, user names and annotations. Note that participants had no access to the ground truth annotations of the test set during the test period.

Table 1 shows the distribution of the three OOV word categories (0, correct; 1, variant; 2, NoES) in both the development corpus and the test corpus. Note that the distribution of the three categories is similar in both corpora. This fact allowed the participants to develop their systems with a corpus that is similar to the test corpus.

[8] `https://dev.twitter.com/terms/api-terms`

[9] `http://komunitatea.elhuyar.org/tweet-norm/iles/2013/06/download_tweets.py`

[10] `http://komunitatea.elhuyar.org/tweet-norm/files/2013/11/tweet-norm_es.zip`

Corpus	#OOV	0	1	2
Development	775	107	600	68
Test	662	98	531	33

Table 1 Distribution of the three OOV word categories (0, correct; 1, variant; 2, NoES) in the development corpus and in the final test corpus.

6 Tweet-Norm shared task

In this section, we first set out to describe the objective of the shared task. Then, we describe the characteristics of the systems that participated in the shared task, and finally present and analyze the results.

6.1 Objective and Evaluation Criteria

The Tweet-Norm shared task aimed at normalizing words unknown to the analyzer at the preprocessing step, such as abbreviations, misspellings, words with repeated characters, etc. Following the line of work of [10] we focus on lexical normalization, whereas other phenomena such as syntactical or stylistic variants are left out of this task.

The goal of the task is to measure how accurate a system is at normalizing OOV words found in tweets. This goal does not involve the classification of the OOV words into different categories (0, 1 and 2, as described in previous section). Instead, the task focuses on identifying whether an OOV word needs to be corrected, and on providing the correct alternative when necessary. Participants had to determine if an OOV word should be deemed correct (e.g., new named entities, words in other language, etc.) or it should be assigned a normalized variation. We measured the accuracy of each system when performing the final evaluation as follows:

- **Correct**: if the OOV word is correct (category 0) or NoES (category 2) and the system does not provide any correction, or if the OOV word is a variant (category 1) and the word suggested by the system to normalize the OOV word is correct.
- **Incorrect**: otherwise.

In order to measure the performance of the systems, we relied on the precision score, defined as the number of correct responses of a system over the whole set of OOV words in the test corpus:

$$P(system_i) = \frac{\text{\#correct suggestions}}{\text{\#OOV words}}. \qquad (1)$$

6.2 Short Description of the Systems

In this section we describe approaches utilized and the characteristics of the systems developed by the participants of the shared task. Next, we list the names of the participants, and describe the systems submitted by them. If it is not stated otherwise, descriptions correspond to the best submitted runs.

RAE [28]: RAE's system devises several rewriting rules that model specific spelling phenomena with very high precision. Other rules include edit distance and typing errors. These rules are compiled into finite state transducers that can be recombined in order to produce a confusion set for OOV words. OOV normalization candidates must occur in a lexicon composed of the DRAE dictionary for Spanish words, the BNC corpus for English words, and a list of NEs compiled from many sources. A language model (LM) decodes the word graph obtained, and determines the most probable sequence for each tweet. The system uses a 3-gram LM obtained from crawling 20k Spanish web pages.

Citius-Imaxin [8]: This system produces two sets of candidates, using several lexical resources (including the DRAE dictionary, a normalization dictionary and names collected from Wikipedia). It also makes use of a set of predefined rules for three kinds of non-standard forms that need to be normalized, i.e., capital letters, repeated characters, and common misspellings. The system is trained with a LM developed from a news corpus gathered from RSS feeds, which is then used to select among the candidates. More precisely, the local context of each candidate is compared, by computing chi-square measure, against the LM which consists of bigrams of tokens found within a window of size 4 (2 tokens to the left and 2 to the right of a given token).

UPC [1]: To produce spelling alternatives for the OOV words, this system searches for similar variants in several gazetteers and lexica (Spanish, English, NEs, morphological derivates of Spanish words) using edit distance measure with several cost matrix: one for keyboard typos, one for phonetic similarity and normal edit distance. It also relies on hand-crafted regular expressions to detect onomatopoeias, acronyms and common shorthands. The final candidate is chosen with a weighed voting scheme: each producer (pair of lexicon and search method) is assigned a weight which is equivalent to its precision on the development data.

Elhuyar [30]: This system first generates all the possible candidates for the OOV words in a tweet, and then selects the combination of candidates that best fits a LM. For the generation of candidates, it combines common abbreviations, colloquial expressions, repeated characters, onomatopoeias and typographical/orthographical errors. Reference lexica of normalized forms were generated from various resources. The LM used for the selection of correct candidates is built using SRILM based on bigrams obtained from Wikipedia articles and a news corpus from EFE[11], a Spanish news agency.

IXA-EHU [2]: This system uses hand-written rules (using *foma*) for the most common phenomena. These rules are incrementally applied. In a complementary way basic orthographic changes are learned and weighed using a noisy-channel model (*Phonetisaurus* tool). Frequencies on the full corpora of tweets after selection of correct words are used as LM. One-to-several correspondences are generated and filtered using a search engine.

[11] `http://www.efe.com/`

Vicomtech [29]: This system performs a first-pass correction using regular expressions and custom lists, which detects and corrects common errors and abbreviations. Then, it uses the edit distance as a measure, up to distance 2, to obtain spelling alternatives. The alternatives are ranked according to a LM and the edit distance scores. A postprocessing step detects NEs and corrects capitalization. The system is fed with a standard dictionary and NE lists obtained from multiple sources. The LM consists of 5-grams of film and documentary captions, although other alternative LMs were also tested.

UniArizona [14]: This system uses contextual phonological replacement rules in the form of transducers to convert the OOV into legitimate lexicon words. Two implementations of the same strategy are given: the first uses hand-crafted transformation rules (about 20), while the second automatically learns this rules using the noisy channel model (combining a transformation model and word frequency). The correction rules range from very specific to highly generic.

UPF-Havas [25]: After separating Twitter metalanguage elements using regular expressions, the OOV words go through a pipeline with feedback loops. This consists of several stages: Spanish dictionary look-up using some spelling variants (case and accents), SMS dictionary look-up, repeated character correction, correction through an open-source spell-checker. The dictionary includes Spanish common names and NEs obtained from Wikipedia articles.

DLSIAlicante [24]: First of all, the system attempts to find the OOV word in the dictionary (aspell dictionary enriched with NEs) after applying heuristic rules to reduce character repetitions, the conversion of numerals into text and a table of abbreviations. If not successful, then the dictionary is indexed using the Metaphone algorithm and the closest spelling alternative is found using the longest common subsequence method. In the case of a tie, a 3-gram LM is used to select the final candidate.

UniMelbourne [12]: In this approach, a collection of 280 million Spanish tweets is used as a source of IV words. For each OOV, the system generates spelling alternatives considering all words in the collection that have small edit distance either orthographically or under the Metaphone phonetic representation. The best correction is selected with a measure of distributional semantics similarity (KL divergence, [18]) in a window of two words. Additionally, a lexicon of slang and abbreviations has been hand-crafted.

UniSevilla [6]: The normalization process begins with filtering the OOV words using a set of lexica covering the Spanish language, as well as small dictionaries built with Twitter vocabulary, emoticons and colloquial inflections. The OOVs are processed with hand-crafted rules (e.g., for character repetition or SMS language), spelling corrections based on edit distance, and a language identifier module. All this information is taken by a candidate selector based on confidence values that produces the final output.

UJaen-Sinai [23]: This system uses a small lexicon of abbreviations and regular expressions that capture onomatopoeias. Then a spell-checker is used to produce spelling alternatives. The spell-checker lexicon is enriched with NEs (from Wikipedia and geographic information sources), popular Twitter jargon, neologisms and interjections in Spanish. The normalized candidate is selected with a unigram LM.

UniCoruña [31]: This system is based on a pipeline that applies rules that detect and normalize onomatopoeias, reduction of character repetitions, diacritic variations and a general purpose spell-checker. The system is trained with a SMS lexicon to enrich the spell-checker dictionary.

6.3 Results

Table 2 shows the accuracy results obtained by the 13 participants[12] in the shared task. The table includes an extra column with a second precision value for participants who submitted two runs. Besides the results of the participants, we also show two more results as references. Firstly, the *Baseline* would be the result of deeming all OOV words correct, therefore without suggesting any changes at all from the input –this would achieve a precision of 0.198. Secondly, the *Oracle* is the aggregated precision value of words that were correctly guessed by at least one of the participants. With a precision of 0.927, only 7.3% of the OOV words were missed by all of the participants.

The system presented by RAE clearly outperformed the rest of the systems, with 18.7% gain over the runner-up, Citius-Imaxin. Most of the other systems achieved intermediate precision values that range from 54% to 67%. We believe that one of the features that stand out from the winners' systems is the careful integration of different components that consider a number of misspelling cases, as well as the quality and coverage of the components utilized. We comment on the results in detail in Section 7.2.

Appendix I shows the list of OOVs that none of the systems has normalized correctly (39 words, 7.25% of the total). This list features a diverse variety of deformations and modifications: for example, the pair *filosofia*/*Filosofía* (Philosophy) requires correcting capitalization and accents at the same time; the pair *yaa*/*allá* (there), although not a standard abbreviation, is orthographically very distant and the word *ya* (already) looks like a much more suitable alternative.

7 Analysis of Results and Discussion

In this Section, we analyze word categories, the components of the systems, and the techniques and resources they used.

7.1 Results by Word Category

Now we delve into the results by breaking down their performance by word category. This allows us to perform a deeper study of the systems' outputs, finding out the categories over which each system performs better.

[12] Out of 20 initially registered participants, 13 groups sent results.

Rank	System	Prec1	Prec2
—	*Oracle*	0.927	—
1	RAE	0.781	—
2	Citius-Imaxin	0.663	0.662
3	UPC	0.653	—
4	Elhuyar	0.636	0.634
5	EHU	0.619	0.609
6	Vicomtech	0.606	—
7	UArizona	0.604	—
8	UPF	0.548	0.491
9	UAlicante	0.545	0.521
10	UMelbourne	0.539	0.517
11	USevilla	0.396	—
12	UJaen	0.376	—
13	UCoruña	0.335	—
—	*Baseline*	0.198	—

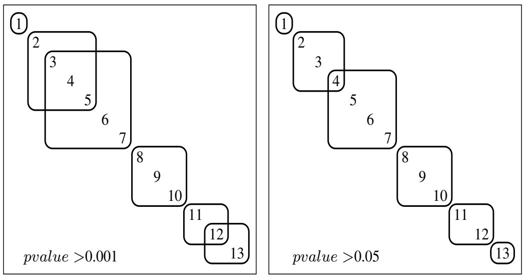


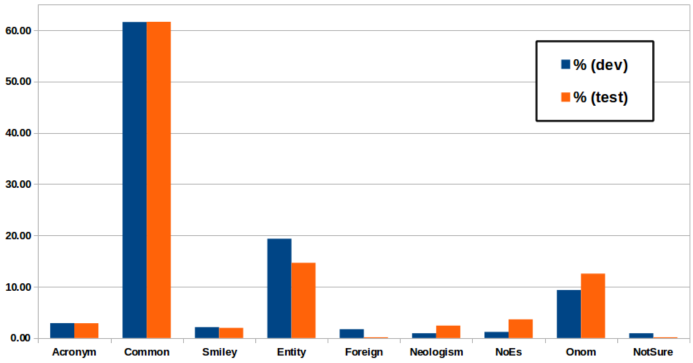
Table 2 Precision of the Tweet-Norm 2013 participants. The graphs on the right side show the results of a statistical significance test using McNemar's test. Two systems (based on Prec1) share a cluster if they are not significantly different under the reported $pvalue$ (either $0,001$ or $0,05$).

First of all, all the OOVs in both the development and test corpora were manually categorized into one of the following word categories: acronym, common word, smiley, entity, foreign word, neologism, not in Spanish (NoEs), onomatopoeia, or unsure. Note that *NoEs* words only include those that are part of a tweet written in Spanish, but for some reason it contains some non-Spanish words; *Unsure* was reserved for cases where either the type was unclear, or none of the predefined categories was suitable. Table 3 shows the distribution of these categories in the corpora. It can be seen that word categories are similarly distributed in the development and test corpora. In both cases, common words, entities, and onomatopoeias are the most frequent word categories, in that order. The other word categories are less popular, and do not even account for 4% of occurrences in any case.

So how did participants do with respect to each word category? Next, we look at the performance of each participating system broken down into word categories. Table 4 shows the precision values by category for the best run for each of the 13 participants, where the rows represent participants and are ordered by overall performance, and the columns represent word categories and are ordered by frequency in the test corpus.

RAE outperformed all the other participants for the three most popular word categories. These are common words, entities, and onomatopoeias, which account for 88.8% of the words in the corpus. The performance gains over the runner-ups for common words, entities, and onomatopoeias, are 21.9%, 19.2% and 8% respectively, which leads to a large extent to the performance gain of 17.8% over the runner-up for the overall performance, *Citius-Imaxin*.

RAE's system was outperformed by at least another system for some word categories of lower frequency. This includes (i) acronyms, where USevilla performed 50% better, (ii) neologisms, where UArizona performed 30% better, (iii) smileys, where as many as 7 systems performed 44.4% better, and (iv) NoEs, where UArizona and UMelbourne's systems performed 11.1% better. Despite the lower frequency of



	Acronym	Common	Smiley	Entity	Foreign	Neologism	NoEs	Onom.	Unsure
Dev	2.87	61.63	1.96	14.65	0.15	2.42	3.63	12.54	0.15
Test	2.89	61.58	2.11	19.34	1.71	0.92	1.18	9.34	0.92

Table 3 Distributions (in percents) of word categories in the development and test corpora.

these word categories, which have little impact on the overall performance, this posits several ways for improving RAE's system. Still, RAE performed better than the average for all word categories.

The most frequent word category (common) accounts for 62% of the OOVs, which is therefore the main factor that determines the final performance of each system. It can be seen that there is a strong correlation between the ranking based on the overall performance, and the ranking based on the performance for common words. There are just a few exceptions at the bottom of the ranking, but the ranking would be the same for the top 6 systems if we only considered common words. For some of the exceptions at the bottom of the ranking, such as UAlicante, USevilla, UJaen, and UCoruña, the performance for entities is substantially lower than their performance for common words and onomatopoeias (as low as 1% in the case of UJaen). The fact that these systems performed very differently for the other frequent categories, i.e., entities and onomatopoeias, explains this difference between the common and total scores. The performance drop for the entity category may occur in part due to the use of their own OOV detection mechanism, as explained in Section 7.3.

The oracle shows the performance of the best possible system by considering all the words that were accurately normalized by at least one participant. With a 0.927 overall performance for the oracle, it shows that only 7.3% of the words were not normalized correctly by any of the participants. However, while common words represent the most frequent category in the corpus, it is also the one with highest percentage of words missed by everyone (15.8%). The second most frequent category, entities, had also 8.1% of the words missed by everyone. This posits the need for further exploring normalization of these two categories, common words and entities, in future research. Improving these would substantially improve the overall performance of normalization systems. On the other hand, a look at the average per-

Rank	System	Com.	Ent.	Ono.	NoEs	Acr.	Neol.	Smil.	Fore.	Unsu.	Total
—	*Oracle*	0.842	0.919	1.000	0.938	1.000	0.870	1.000	1.000	0.952	0.927
1	RAE	**0.806**	**0.897**	**0.651**	0.750	0.421	0.625	0.692	**1.000**	**1.000**	0.781
2	Citius-Imaxin	0.662	0.753	0.554	0.750	0.474	0.563	**1.000**	**1.000**	0.000	0.663
3	UPC	0.652	0.701	0.602	0.667	0.368	0.625	**1.000**	**1.000**	**1.000**	0.653
4	Elhuyar	0.630	0.711	0.542	0.750	0.526	0.438	**1.000**	**1.000**	**1.000**	0.636
5	EHU	0.625	0.649	0.530	0.667	0.368	0.688	**1.000**	**1.000**	0.000	0.619
6	Vicomtech	0.610	0.670	0.530	0.417	0.579	0.438	**1.000**	**1.000**	**1.000**	0.606
7	UArizona	0.588	0.598	0.530	**0.833**	0.579	**0.813**	**1.000**	**1.000**	0.000	0.604
8	UPF	0.576	0.649	0.578	0.292	0.211	0.250	0.154	0.000	0.000	0.548
9	UAlicante	0.598	0.433	0.590	0.292	0.421	0.438	0.154	**1.000**	**1.000**	0.545
10	UMelbourne	0.471	0.732	0.434	**0.833**	0.526	0.750	**1.000**	**1.000**	**1.000**	0.538
11	USevilla	0.407	0.258	0.349	0.417	**0.632**	0.375	0.923	**1.000**	**1.000**	0.396
12	UJaen	0.502	0.010	0.494	0.000	0.000	0.125	0.000	0.000	0.000	0.376
13	UCoruña	0.456	0.041	0.373	0.042	0.000	0.000	0.000	0.000	0.000	0.335
—	*Baseline*	0.526	0.032	1.000	0.557	1.000	0.750	0.917	1.000	0.049	0.196
—	*Average*	0.583	0.546	0.520	0.516	0.393	0.471	0.686	0.769	0.538	0.562
—	*Best*	0.806	0.897	0.651	0.833	0.632	0.813	1.000	1.000	1.000	0.781

Table 4 Precision values broken down into word categories for the best run for each of the participants, and average and best performance for each word category. Participants in rows are ordered by overall performance, while word categories in columns are ordered by their frequency in the test corpus. Bold figures represent participants who obtained the highest precision score for the word category in question.

formance shows that acronyms were the most difficult overall, and certainly a word category that needs careful analysis for improvement in future work.

7.2 Focused phenomena

The good performance of the RAE system is remarkable. It for the most part outperforms all the others with a 78% score for precision, while most systems score between 54% and 67%. The difference can be explained by the thorough and detailed treatment of many linguistic phenomena appearing in Twitter, the statistical combination of the used modules, and the quality and coverage of used resources. Note that Han et al. (2013) describe a system for English that achieves 72.3% F-score in a similar scenario to that described in our proposal: OOV detection + normalization. Only the first Spanish system in the competition outperforms this score. The main diffence with regard to our evaluation protocol is that we use one-to-many correction pairs, while Han et al. (2013) only use one-to-one pairs.

The phenomena explicitly addressed by several of the participant systems are the following:

- Usual orthographic mistakes (*h* →).
- Usual phonological changes (*c/qu* → *k*).
- Omission of graphical accent (*á* → *a*).
- Omission of characters, mainly vowels and final letters, especially in participles (*encantado* → *encantao*).
- Use of abbreviations or reduction of words to their initial characters (*examen* → *exam*)

- Emphasis expressed via character repetition (usually vowels) (*felicidades* → *felicidadeeees*).
- Omitted capitalization (*Juan* → *juan*).
- Contiguous word joining *es que* → *esque*.
- Logograms and pictograms. (*por* → *x*; *dos* → *2*).
- Repetition of onomatopoeias (*ja* → *jajajaja*).

The lexica used by the participants to look for normalized variations are mostly Spanish dictionaries, spell checkers, and also Freeling —i.e., the same tool used for the preprocessing step. Some have also used other resources: (i) English dictionaries to look for OOV words that, without being Spanish words, do not need to be changed, (ii) the Spanish Wikipedia[13] to identify named entities, (iii) small slang and variation dictionaries, and (iv) word frequencies extracted from other corpora to identify common misspellings on the Internet and Twitter.

Different approaches that make use of language models have also relied on several corpora of Spanish language texts. Both general purpose corpora and specific Twitter corpora have been used to create language models. One of the participating systems used the API of a search engine to filter multi-word terms.

The participants also utilized several tools to create their normalization systems. Many used spell checkers (e.g., Aspell,[14] Hunspell,[15] Jazzy[16]), which can also be used to look for alternative candidates. Some also used Foma[17] to work with transducers, which learns transformation rules for phonemes and graphemes. In some cases, transformation rules have also been defined based on language models, e.g., using Phonetisaurus.[18] For the selection of the final candidate, some relied on corpora-based frequencies, whereas others used language modeling tools (e.g., OpenGrm[19] and SRILM[20]).

7.3 Summary of Techniques and Resources

Table 5 summarizes the characteristics of each system participating in the Tweet-Norm 2013 evaluation. These characteristics include only the best run from each participant. The table is divided into four parts according to the clusters obtained in Table 2 for a $pvalue > 0.001$: 1, 2–7, 8–10, 11–13. There are eight columns containing the analyzed characteristics for each system. Their meaning is as follows:

1. Architecture: **G/F**: Generate/Filter architecture. A generation process proposes a set of alternative spellings (IV words) for each OOV, this is called the *confusion set*. In a second step a filtering mechanism is implemented to select one of the

[13] http://es.wikipedia.org

[14] http://aspell.net

[15] http://hunspell.sourceforge.net

[16] http://jazzy.sourceforge.net

[17] https://code.google.com/p/foma/

[18] http://code.google.com/p/phonetisaurus/

[19] http://www.opengrm.org

[20] http://www.speech.sri.com/projects/srilm/

Rank	System	Architecture	Filtering	LM	Rules	Ap. Search	SMS	Phonetics	NEs
1	RAE	G/F	LM	3-gram	R	ED	–	PH	E
2	Citius-Imaxin	G/F	LM	2-gram	R	ED	–	–	E
3	UPC	G/F	S(vot)	–	R	ED	–	PH	E
4	Elhuyar	G/F	LM	3-gram	R	LCS	SMS	–	E
5	EHU	PP	–	1/2-gram	R	–	–	–	E
6	Vicomtech	G/F	LM+S	5-gram	–	ED	–	–	E
7	UArizona	PP	–	–	R	–	–	–	–
8	UPF	PP	–	–	–	Spell	SMS	–	I
9	UAlicante†	G/F	LM	3-gram	R	LCS	–	MPH	I
10	UMelbourne	G/F	S(dist)	–	–	ED	SMS	MPH	I
11	USevilla†	G/F	S(conf)	–	R	ED	–	–	–
12	UJaen	G/F	F	–	R	Spell	SMS	–	E
13	UCoruña†	G/F	S	–	R	Spell	SMS	–	–

Table 5 Synoptic table of system's charecteristics. See Section 7.3 for details.

proposed words in the confusion set. This architecture is used by 10 out of the 13 systems. **PP**: Pipeline architecture. Each OOV word goes through a sequence of analysers. The process stops when an IV word is produced by one of the analyzers.

2. Filtering Mechanisms (for G/F architectures): **LM**: a Language Model selects the most probable candidate from the confusion set according to the context words. **S**: some Scoring function is derived from the generation of IV words. This may implement voting or other confidence estimation techniques such as distributional similarity. **F**: The most frequent word in a sample corpus is selected. It is a particular case of LM of length 1.
3. Language Model: n**-gram**: A language model of length n is used (not necessary as a Filtering mechanism).
4. Rules: **R**: Some kind of knowledge-based transformation rules are implemented (e.g., shorthands, phonographemes). The actual number and complexity of the rules may be disparate.
5. Approximated Search: **ED**: Edit Distance is used to find similar IV words. **LCS**: Longest Common Subsequence is used to find similar IV words. **Spell**: A spell-checker is used to find similar IV words.
6. SMS: **SMS**: The system uses dictionaries of textese utilized in SMS, and slang.
7. Phonetics: **MPH**: Phonetic representations of words are obtained with the Metaphone algorithm. **PH**: Words are represented with IPA phonemes.
8. Named Entities: **E**: Explicit lists of NEs are compiled from one or more sources as new IV words. **I**: The dictionary of IV words is enriched with textual sources thus, NEs are implicitly added.

A dash (i.e. –) is used for the systems that do not have that feature. Finally, the dagger (i.e. †) notes that the system uses its own OOV detection mechanism instead of the ones provided in the test set. In this case, we have to be cautious when drawing comparisons with the rest of the systems.

7.4 Discussion

According to the clusters obtained in Table 2 for a $pvalue > 0.001$, we can divide the systems in four groups in this way: 1, 2–7, 8–10, 11–13. Looking at the columns of Table 5, we can characterize the systems within each group as follows:

- Five out of the top seven systems use a generate/filter architecture with a language model as filter, while in the lower half of the table many systems use local or confidence based scoring mechanisms. This suggests that a filter operating on a OOV's confusion set cannot work solely on the intrinsic properties of the words but on its context.
- Using a language model seems to be a competitive way of scoring and selecting a good normalization in the generate/filter architecture, and it is consistently better than other scoring methods devised by the participants.
- The top six systems have compiled extensive lists of named entities, while the rest have not targeted this kind of knowledge or have done so indirectly. Surprisingly, the precision for entity normalization does not seem to be explained by this feature, neither by the use of SMS lexica.
- Using the phonetic representation of words seems to be positive although not as much when the Metaphone algorithm is used. This suggests that Metaphone may not be a suitable method for generating alternatives. Unfortunately, the performance of the two systems using Metaphone (UAlicante, UMelbourne) varies greatly in the three most frequent categories.
- One reason that explains the good results of the RAE system is that, compared to others, their mechanism for confusion set generation seems very precise [28] and very little noise has to be filtered out afterwards. For other systems that, like UPC, can generate thousands of alternative spellings [1], it becomes much more difficult to select the correct candidate. EHU and UArizona rely mainly on a sound set of transformation rules (hand-crafted for UArizona) and also achieve good results with a pipeline architecture.
- UMelbourne is especially good in the NoEs and neologism categories. Their approach based on distributional semantics on a large corpus of tweets may explain their good results. Under the hypothesis that the use of neologisms and foreign words in Twitter is some kind of slang (i.e., slang replaces well-known words, it is informal, the users are familiar with its context of application), the vocabulary size of these categories is probably much reduced when compared to common words. Therefore, their context of usage may be more accurately characterized by distributional semantics. This reasoning can be applied to some extent to the entity category, in which UMelbourne has notable results too.

8 Conclusions and future work

The development of the benchmark evaluation framework and the *TweetNorm_es* corpus, as well as the Tweet-Norm 2013 shared task that enabled evaluation of systems from 13 participants, served as an initial step toward encouraging implementation of new methods for and approaches to Spanish microtext normalization in the research community. The high number of participants has proven the task relevant, and posited a number of issues to be considered in future research.

The work presented in this paper paves the way for future development and research on Spanish microtext normalization, setting forth a methodology to create a corpus for these purposes, as well as releasing the corpus we created following such methodology. The corpus provides a gold-standard for development and evaluation of microtext normalization tools.

The corpus is available under the terms of the CC-BY license for anyone interested in the task, and can be found at the website of the workshop.[21]

This work has also brought to light a number of issues that remain unresolved and are worth studying in future work. Here we have performed *in vitro* evaluations of the normalization systems. We believe that *in vivo* evaluations by incorporating normalization into other NLP systems, such as sentiment analysis or machine translation will enable deeper study of the task, as well as to quantify the actual effect of processing normalized outputs. Additionally, we would like to broaden the task by not only dealing with lexical normalization, but also addressing complementary tasks such as normalization of syntax and/or real-word errors. Last but not least, we are aware that the size of the corpus is limited. Extending the corpus and considering different OOV categories would enable to perform a more detailed evaluation, especially for machine learning purposes.

Acknowledgments

We would like to thank all the members of the organizing committee. This work has been supported by the following projects: Spanish MICINN projects *Tacardi* (Grant No. TIN2012-38523-C02-01), *Skater* (Grant No. TIN2012-38584-C06-01), *TextMESS2* (TIN2009-13391-C04-01), *OntoPedia* (Grant No. FFI2010-14986) and *Holopedia* (TIN2010-21128-C02-01); *Xlike* FP7 project (Grant No. FP7-ICT-2011.4.2-288342); UNED project (2012V/PUNED/0004); *ENEUS-Marie Curie Actions* (FP7/2012-2014 under REA grant agreement n°302038); *Celtic* CDTI FEDER-INNTER-CONECTA project (Grant No. ITC-20113031); Research Network MA2VICMR (S-2009 / TIC-1542); and *HPCPLN* (Grant No. EM13/041, Xunta de Galicia).

References

1. Ageno, A., Comas, P.R., Padró, L., Turmo, J.: The talp-upc approach to tweet-norm 2013. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
2. Alegria, I., Etxeberria, I., Labaka, G.: Una cascada de transductores simples para normalizar tweets. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
3. Beaufort, R., Roekhaut, S., Cougnon, L.A., Fairon, C.: A hybrid rule/model-based finite-state framework for normalizing SMS messages. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 770–779. Uppsala, Sweden (2010)

[21] http://komunitatea.elhuyar.org/tweet-norm/

4. Chakrabarti, D., Punera, K.: Event summarization using tweets. In: Proceedings of the Fifth International Conference on Weblogs and Social Media (ICWSM) (2011)
5. Costa-Jussà, M.R., Banchs, R.E.: Automatic normalization of short texts by combining statistical and rule-based techniques. Language Resources and Evaluation pp. 1–15 (2013)
6. Cotelo-Moya, J.M., Cruz, F.L., Troyano, J.A.: Resource-based lexical approach to tweet-norm task. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
7. Eisenstein, J.: What to do about bad language on the internet. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 359–369 (2013)
8. Gamallo, P., Garcia, M., Pichel, J.R.: A method to lexical normalisation of tweets. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
9. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford pp. 1–12 (2009)
10. Han, B., Baldwin, T.: Lexical normalisation of short text messages: Makn sens a #twitter. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 368–378 (2011)
11. Han, B., Cook, P., Baldwin, T.: Lexical normalisation for social media text. ACM Transactions on Intelligent Systems and Technology (TIST) **43**(1), 15–27 (2013)
12. Han, B., Cook, P., Baldwin, T.: unimelb: Spanish text normalisation. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
13. Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the First Workshop on Social Media Analytics, pp. 80–88. ACM (2010)
14. Hulden, M., Francom, J.: Weighted and unweighted transducers for tweet normalization. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
15. Inouye, D., Kalita, J.K.: Comparing twitter summarization algorithms for multiple post summaries. In: Proceedings of the IEEE Third International Conference on Social Computing (SocialCom), pp. 298–306. IEEE (2011)
16. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 151–160 (2011)
17. Kaufmann, J., Kalita, J.: Syntactic normalization of twitter messages. In: Proceedings of the International Conference on Natural Language Processing. Kharagpur, India (2010)
18. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics **22**(1), 79–86 (1951)
19. Lin, J., Snow, R., Morgan, W.: Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD), pp. 422–429. ACM (2011)
20. Ling, W., Dyer, C., Black, A.W., Trancoso, I.: Paraphrasing 4 microblog normalization. In: Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP), pp. 73–84 (2013)
21. Liu, F., Weng, F., Jiang, X.: A broad-coverage normalization system for social media language. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, pp. 1035–1044. Association for Computational Linguistics (2012)
22. Liu, X., Wei, F., Zhang, S., Zhou, M.: Named entity recognition for tweets. ACM Transactions on Intelligent Systems and Technology (TIST) **4**(1), 3 (2013)
23. Montejo-Ráez, A., Díaz-Galiano, M., Martínez-Cámara, E., Martín-Valdivia, T., García-Cumbreras, M.A., Ureña-López, A.: Sinai at twitter-normalization 2013. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
24. Mosquera-López, A., Moreda, P.: Dlsi en tweet-norm 2013: Normalización de tweets en español. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
25. Muñoz-García, O., Suárez, S.V., Bel, N.: Exploiting web-based collective knowledge for micropost normalisation. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)

26. Oliva, J., Serrano, J.I., del Castillo, M.D., Iglesias, Á.: A SMS normalization system integrating multiple grammatical resources. Natural Language Engineering **19**(1), 121–141 (2013)
27. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC) (2012)
28. Porta, J., Sancho, J.L.: Word normalization in twitter using finite-state transducers. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
29. Ruiz, P., Cuadros, M., Etchegoyhen, T.: Lexical normalization of spanish tweets with preprocessing rules, domain-specific edit distances, and language models. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
30. Saralegi, X., San-Vicente, I.: Elhuyar at tweet-norm 2013. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
31. Vilares, J., Alonso, M.A., Vilares, D.: Prototipado rápido de un sistema de normalización de tuits: Una aproximación léxica. In: Proceedings of the Tweet Normalization Workshop at the Conference of the Spanish Society for Natural Language Processing (SEPLN) (2013)
32. Villena Román, J., Lana Serrano, S., Martínez Cámara, E., González Cristóbal, J.C.: TASS-workshop on sentiment analysis at SEPLN. Proceedings of the Spanish Society for Natural Language Processing (SEPLN) (2013)
33. Wang, A., Kan, M.Y., Andrade, D., Onishi, T., Ishikawa, K.: Chinese informal word normalization: an experimental study. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP), vol. 13, pp. 127–135 (2013)
34. Wei, Z., Zhou, L., Li, B., Wong, K.F., Gao, W., Wong, K.F.: Exploring tweets normalization and query time sensitivity for twitter search. In: Proceedings of the Text REtrieval Conference (TREC) (2011)

Appendix I: List of unresolved OOV words

Table 8 contains the list of words from the corpus that none of the systems found the correct variation for. The list comprises the word as spelled originally in the corpus on the left column, and the correct variation annotated manually on the right column.

Original	Variation
FYQ	Física_y_química
sisiii	sí_sí
yaa	allá
picolos	picoletos
nainonainonahh	nainonainoná
gordys	gorditas
JUUUM	hum
Tuitutil	TuitÚtil
crst	Cristo
mencantaba	me_encantaba
diitaas	diítas
soo	eso
queeee	qué
Teinfiniteamo	Te_amo_infinitamente
aber	a_ver
Hum	Humedad
L.	l.
Muchomuchacho	Mucho_Muchacho
Hojo	Jo
jonaticas	jonáticas
gafis	gafitas
her	hermano|hermana
MIAMOR	mi_amor
guapii	guapita
WAPAHHH	guapa
EAEA	ea_ea
Acho	Macho
tirantitas	tirantitos
HMYV	MHYV
filosofia	Filosofía
nah	nada
FAV	favorito
JIIIIIIIIOLE	Olé
Fotazo	fotaza
gor	gorda|gordo
coner	con_el
shh	sí|sé
primera+	primera_más
salobreja	Salobreja

Table 6 List of OOV words for which none of the participants found the correct variation.