Exploiting web-based collective knowledge for micropost normalisation

Uso del conocimiento colectivo recogido en recursos de la Web para la normalización de textos cortos publicados en Twitter

Óscar Muñoz-García
Havas Media Group
Madrid - Spain
oscar.munoz@havasmg.com

Silvia Vázquez, Nuria Bel
Universitat Pompeu Fabra
Barcelona - Spain
silvia.vazquez@upf.edu, nuria.bel@upf.edu

Resumen: La tarea de normalización de contenido publicado por el usuario es un paso fundamental previo al análisis de las publicaciones en los medios sociales, especialmente en Twitter. En este trabajo se presenta un método para la normalización morfológica de *tweets* mediante el uso de recursos publicados en la Web y desarrollados de manera colectiva, entre los que se encuentran la Wikipedia y un diccionario de SMS. Los resultados obtenidos demuestran que estos recursos son una fuente de conocimiento muy valiosa para la generación de los diccionarios utilizados en la tarea de normalización.
Palabras clave: medios sociales, normalización de contenidos, Twitter, tweet-norm

Abstract: The task of normalising user-generated content is a crucial step before analysing social media posts, particularly on Twitter. This paper presents a method for the morphological of tweets by the use of on-line and collectively developed resources, including Wikipedia and a SMS lexicon. The results obtained demonstrate that these resources are a valuable source of knowledge for generating the dictionaries used in the normalisation task.
Keywords: social media, micropost normalisation, Twitter, tweet-norm

1 Introduction and objectives

Microposts published on social media are characterised by informality, brevity, frequent grammatical errors and misspellings, and by the use of abbreviations, acronyms, and emoticons. These features add additional difficulties in text mining processes that frequently make use tools designed for dealing with texts which conform to the canons of standard grammar and spelling (Hovi et al., 2013).

The micropost normalisation task enhances the accuracy of NLP tools when applied to short fragments of texts published in social media, e.g., the syntactic normalisation of tweets may improve the accuracy of existing part-of-speech taggers (Codina and Atserias, 2012).

The collective knowledge freely available on the Web, and particularly Wikipedia, has been used in different NLP tasks, such as text categorization (Gabrilovich and Markovitch, 2006), topic identification (Coursey, Mihalcea, and Moen, 2009), measuring the semantic similarity between texts (Gabrilovich and Markovitch, 2007), and word sense disambiguation (Mihalcea, 2007) among others.

This paper presents a technique for morphological normalisation of microposts by the use of two open data sources namely, Wikipedia and the SMS dictionary of the Spanish Association of Internet Users (AUI, 2013).

The paper is structured as follows. Section 2 describes the architecture and the components of the system. Section 3 describes the linguistic resources that we have reused for constructing the normalisation tool. Section 4 presents the evaluation results. Finally, Section 5 presents the conclusions and future lines of work.

2 Architecture and components of the system

Figure 1 shows the process followed by the micropost normaliser proposed. The specific components involved in the overall process are described below.

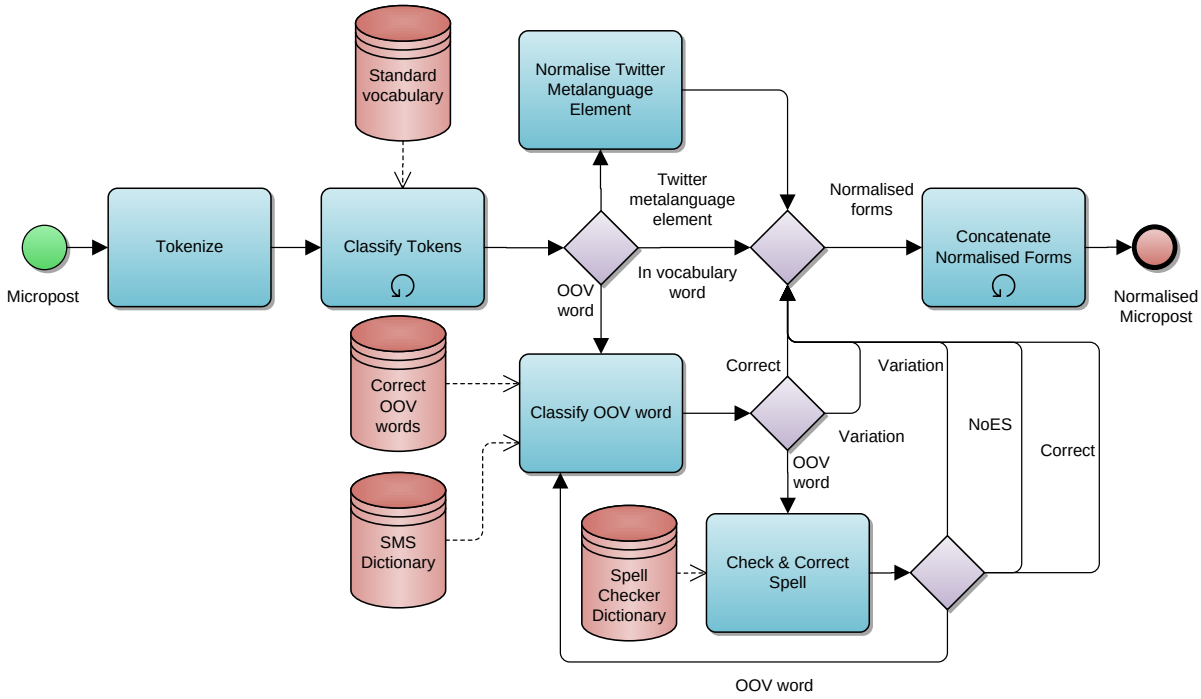

Figure 1: Normalisation process

2.1 Tokeniser

This component receives the text to be normalised and breaks it into words, Twitter metalanguage elements (e.g., hash-tags, user IDs), emoticons, URLs etc. The output (i.e., the list of tokens) is sent to the *Token Classifier* component.

2.2 Token Classifier

The input of this component is the list of tokens generated by the *Tokeniser*. It classifies each of them into one of the following categories:

- Twitter metalanguage elements (i.e., hash-tags, user IDs, RTs and URLs). Such elements are detected by matching regular expressions against the token (e.g., if a token starts by the symbol "#", then it is a hash-tag). Each token classified in this category is sent to the *Twitter Metalanguage Normaliser* component.
- Words contained in a standard language dictionary, excluding proper nouns. Each token classified in this category is sent to the *Normalised Forms Concatenator* component.
- Out-Of-Vocabulary (OOV) words. They are words not found in a standard dictionary, neither are Twitter metalanguage elements. Each token classified in this category is sent to the *OOV Word Classifier* component.

2.3 OOV Word Classifier

This component receives every token previously classified as OOV by the *Token Classifier* and detects if it is correct, wrong, or unknown. If the token is wrong, the component returns the correct form of the token. The *OOV Word Classifier Component* executes the following process:

1. Firstly, the token is looked up in a dictionary of correct OOV words. The search disregards both case and accents.
   (a) If an exact match of the token is found in the dictionary (e.g., both forms are capitalised), then the token is classified as *Correct* and sent to the *Normalised Forms Concatenator* component with no variation.
   (b) If the token is found with variations of case or accentuation, then the token is classified as *Variation* and its correct form is sent to *Normalised Forms Concatenator* component.

(c) If the token is not found in the dictionary, then the process continues in step 2.

2. The token is looked up in a SMS dictionary which contains tuples with the SMS term and its corresponding correct form. The search is case-unsensitive, and does not consider accent marks.

   (a) If the token is found in the SMS dictionary, then it is classified as *Variation* and its correct form is retrieved and sent to *Normalised Forms Concatenator* component.

   (b) If the token is not found in the dictionary, then it is sent to the *Spell Checker and Corrector* component.

2.4 Spell Checker and Corrector

This component checks the spelling of the token received and returns its correct form when possible. To do so, it executes the following process:

1. Firstly, the token is matched against regular expressions to find whether it contains characters (or sequences of characters) repeated more than twice (e.g., "looooolllll" and "jajaja").

   (a) If the token contains repeated characters (or sequences of characters), the repeated ones are removed (e.g., "lol", and "ja"), and the resulting form is sent back to the *OOV Word Classifier*, since the new form may be included into the correct words set.

   (b) If the token does not contain repeated characters (or sequences of characters), then the process continues in step 2.

2. The token is sent to an existing spell checking and correction implementation reused by this component.

   (a) If the spell is correct, the token is classified as *Correct* and sent to the *Normalised Forms Concatenator* component without a variation.

   (b) If the spell is not correct, the token is classified as *Variation*, and the first correct form returned by the spelling corrector is sent to *Normalised Forms Concatenator*.

   (c) If the spell checker is not able to propose a correct form, the token is classified as *Unknown* and sent to the *Normalised Forms Concatenator* without a variation.

2.5 Twitter Metalanguage Normaliser

This component performs a syntactic normalisation of Twitter meta-language elements. Specifically, it executes a set of rules, previously proposed by (Kaufmann and Jugal, 2010).

(1) Remove the sequence of characters "RT" followed by a mention to a Twitter user (marked by the symbol "@") and, optionally, by a colon punctuation mark; (2) Remove user IDs that are not preceded by a coordinating or subordinating conjunction, a preposition, or a verb; (3) Remove the word "via" followed by a user mention at the end of the tweet; (4) Remove all the hash-tags found at the end of the tweet; (5) Remove all the "#" symbol from the hash-tags that are maintained; (6) Remove all the hyper-links contained within the tweet; (7) Remove ellipses points that are at the end of the tweet, followed by a hyper-link; (8) Replace underscores with blank spaces; (9) Divide camel-cased words in multiple words (e.g., "BarackObama" is converted to "Barack Obama").

2.6 Normalised Forms Concatenator

This component receives the normalised form of each token, and amends the micropost.

3 Resources employed

The system described makes use of the following resources.

We use Freeling (Padró and Stanilovsky, 2012) for microposts tokenisation. Its specific tokenization rules and its user map module were adapted for dealing with smileys and particular elements typically used in Twitter, such as hash-tags, RTs, and user IDs.

In addition, we use the POS-tagging module of Freeling within the *Token Classifier* component. As we deactivate Freeling's probability assignment and unknown word guesser module, all the words which are not contained in Freeling's POS-tagging dictionary are not marked with a tag and considered

as OOV words. Our standard vocabulary is, thus, the Freeling dictionary itself.

We have populated the correct OOV words dictionary (used by the *OOV Word Classifier* component) by making use of the list of articles' titles from Wikipedia (Wikipedia, 2013). To speed-up the process of querying the 2,447,932 Wikipedia articles' titles, we uploaded them to a HBASE store (Apache, 2013).

In order to increase the coverage of the correct OOV words dictionary, we incorporated into it a list of first names from the Spanish National Institute of Statistics (INE, 2013). This list contains 18,679 male names and 19,817 female names.

Additionally, we have populated the SMS dictionary and its corresponding correct forms, from the SMS dictionary of the Spanish Association of Internet Users (AUI, 2013), which contains 53,281 entries for Spanish.

Finally, the *Spell Checker and Corrector* component makes use of Jazzy (Jazzy, 2013), an open-source Java library. For the creation of the spell checker dictionary used by Jazzy, we made use of the Spanish and Mexican dictionaries available on JazzyDicts (JazzyDicts, 2013). The resulting dictionary contains 683,436 terms.

4 Settings and evaluation

The evaluation of the technique previously described was done by using two development corpora and a test corpus provided by the organisation of the Tweet Normalisation Workshop at SEPLN 2013. Specifically, we evaluated the performance of the OOV identification, classification and correction tasks. The accuracy of the normalization task for the Twitter metalanguage elements was not evaluated since it was out of the scope of the workshop challenge.

Table 1 shows the results of the evaluation, including the size of each evaluation corpus (column 2), the precision obtained by using either Wikipedia or the SMS dictionary separately (columns 3 and 4 respectively), and the overall precision achieved by exploiting both dictionaries (column 5).

As Table 1 reflects, both dictionaries help to improve the final precision score, being the SMS dictionary the one which contributes the most. This can be explained with the coverage of OOV words by each of the dictionaries, which is shown in Table 2. The SMS dictionary contains a bigger percentage of OOV words than the dictionary populated with Wikipedia titles.

Corpus	Size	Wikipedia	SMS	Both
Devel. 1	100	0.336	0.631	0.688
Devel. 2	500	0.317	0.634	0.66
Test	600	0.361	0.516	0.548

Table 1: Precision of the normalisation tool

Corpus	Wikipedia	SMS
Development 1	20.661%	47.107%
Development 2	20.436%	51.188%
Test	27.497%	28.115%

Table 2: Coverage of OOV words by dictionary

5 Conclusions and future work

We presented a method for tweet normalisation that relies on existing web resources collectively developed, finding that such resources, useful for many NLP tasks, are also valid for the task of micropost normalisation.

With respect to the future lines of work, we plan to adapt the normaliser to new languages by the incorporation of the corresponding dictionaries and improving the existing lexicons by the use of more available resources, such as the anchor texts from intra wiki links.

Additionally, we plan to improve the normalization of multiword expressions, as different words should be transformed in just one (e.g., "a cerca de" should be transformed into "acerca de"), as well as cases where joined words should be splitted (e.g. "realmadrid") by using existing word breaking techniques, such as the one described in (Wang, Thraser, and Hsu, 2011).

Finally, we will study how the normalisation process affects to different opinion mining tasks, including sentiment analysis and topic identification.

Acknowledgements

This research is partially supported by the Spanish Centre for the Development of Industrial Technology under the CENIT program, project CEN-20101037, "Social Media" (`http://www.cenitsocialmedia.es`). We are very grateful to AUI (Asociación de Usuarios de Internet) for facilitating the textese dictionary used in this work to us.

References

Apache. 2013. HBase. `http://hbase.apache.org`. [Online; accessed 25-Jul-2013].

AUI. 2013. Asociación de Usuarios de Internet. `http://aui.es`. [Online; accessed 24-July-2013].

Codina, Joan and Jordi Atserias. 2012. What is the text of a tweet? In *Proceedings of @NLP can u tag #user_generated_content?! via lrec-conf.org*, Istanbul, Turkey, May. ELRA.

Coursey, K., R. Mihalcea, and W. Moen. 2009. Using encyclopedic knowledge for automatic topic identification. In *Proc. of the Thirteenth Conference on Computational Natural Language Learning*, pages 210–218. Association for Computational Linguistics.

Gabrilovich, E. and S. Markovitch. 2006. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proc. of the 21st National Conference on Artificial Intelligence*, volume 2, page 1301. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Gabrilovich, E. and S. Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of the 20th Int. Joint Conference on Artificial Intelligence*, pages 6–12.

Hovi, Eduard, Vita Markman, Craig Martell, and David Uthus. 2013. Analyzing microtext. In *Papers from the 2013 AAAI Spring Symposium*. Association for the Advancement of Artificial Intelligence, March.

INE. 2013. INEbase: Operaciones estadísticas: clasificación por temas. `http://www.ine.es/inebmenu/indice.htm`. [Online; accessed 8-April-2013].

Jazzy. 2013. Jazzy. `http://jazzy.sourceforge.net`. [Online; accessed 25-Jul-2013].

JazzyDicts. 2013. JazzyDicts. `http://sourceforge.net/projects/jazzydicts`. [Online; accessed 25-Jul-2013].

Kaufmann, Max and Kalita Jugal. 2010. Syntactic normalization of twitter messages. In *Proceedings of the International Conference on Natural Language Processing (ICON-2010)*.

Mihalcea, R. 2007. Using wikipedia for automatic word sense disambiguation. In *Proc. of NAACL HLT*, volume 2007.

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

Wang, Kuansan, Christopher Thraser, and Paul Bo-June Hsu. 2011. Web Scale NLP: A Case Study on URL Word Breaking. In *Proceedings of the 20th international conference on World Wide Web*, pages 357–366. ACM.

Wikipedia. 2013. Wikipedia:Database download. `http://en.wikipedia.org/wiki/Wikipedia:Database_download`. [Online; accessed 23-May-2013].