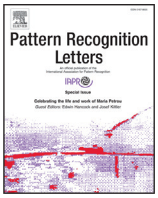
Pattern Recognition Letters 94 (2017) 68–74

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



A simple approach to multilingual polarity classification in Twitter

Eric S. Tellez[a,c], Sabino Miranda-Jiménez[a,c,*], Mario Graff[a,c], Daniela Moctezuma[a,b], Ranyart R. Suárez[d], Oscar S. Siordia[b]

[a] CONACyT Consejo Nacional de Ciencia y Tecnología, Dirección de Cátedras, Insurgentes Sur 1582, Crédito Constructor, 03940, Ciudad de México, México
[b] Centro de Investigación en Geografía y Geomática "Ing. Jorge L. Tamayo", A.C. Circuito Tecnopolo Norte 117, Tecnopolo Pocitos II, 20313, Aguascalientes, México
[c] INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Circuito Tecnopolo Sur 112, Tecnopolo Pocitos II, 20313, Aguascalientes, México
[d] División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica, Universidad Michoacana de San Nicolás de Hidalgo, Santiago Tapia 403, 58000, Morelia, México

ARTICLE INFO

Article history:
Received 24 August 2016
Available online 22 May 2017

Keywords:
Multilingual sentiment analysis
Error-robust text representations
Opinion mining

ABSTRACT

Recently, sentiment analysis has received a lot of attention due to the interest in mining opinions of social media users. Sentiment analysis consists in determining the polarity of a given text, i.e., its degree of positiveness or negativeness. Traditionally, Sentiment Analysis algorithms have been tailored to a specific language given the complexity of having a number of lexical variations and errors introduced by the people generating content. In this contribution, our aim is to provide a simple to implement and easy to use multilingual framework, that can serve as a baseline for sentiment analysis contests, and as a starting point to build new sentiment analysis systems. We compare our approach in eight different languages, three of them correspond to important international contests, namely, SemEval (English), TASS (Spanish), and SENTIPOLC (Italian). Within the competitions, our approach reaches from medium to high positions in the rankings; whereas in the remaining languages our approach outperforms the reported results.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Sentiment analysis is a crucial task in opinion mining field where the goal is to extract opinions, emotions, or attitudes to different entities (person, objects, news, among others). Clearly, this task is of interest for all languages; however, there exists a significant gap between English state-of-the-art methods and other languages. As expected some researchers decide to test the straightforward approach which consists in translating the messages to English, and then, use a high performing English sentiment classifier (for instance, see [3] and [4]), instead of creating a sentiment classifier optimized for a given language. However, the advantages of a properly tuned sentiment classifier have been studied for different languages (see, for instance [1,2,18,25]).

This manuscript focuses on the particular case of multilingual sentiment analysis of short informal texts such as Twitter messages. Our aim is to provide an easy-to-use tool to create sentiment classifiers based on supervised learning (i.e., labeled dataset); where the classifier should be competitive to those sentiment classifiers carefully tuned to a particular language. Furthermore, our second contribution is to create a well-performing baseline to compare new sentiment classifiers in a broad range of languages or to bootstrap new sentiment analysis systems. Our approach is based on selecting, using a search algorithm, a suitable combination of text-transforming techniques commonly used in Information Retrieval and Natural Language Processing such as n-grams of words and q-grams of characters, among others. The goal is that the text transformations selected optimize some performance measure, and the techniques chosen are robust to typical writing errors.

In this context, we propose a robust multilingual sentiment analysis method, tested in eight different languages: Spanish, English, Italian, Arabic, German, Portuguese, Russian and Swedish. We compare the performance of our approach in three international contests: TASS'15, SemEval'15-16 and SENTIPOLC'14, for Spanish, English and Italian respectively; the remaining languages are compared directly with the results reported in the literature. The experimental results locate our approach in good positions for all considered competitions; and excellent results in the other five

[*] Corresponding author at: INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Circuito Tecnopolo Sur No 112, Fracc. Tecnopolo Pocitos II, Aguascalientes 20313, México .

E-mail addresses: eric.tellez@infotec.mx (E.S. Tellez), sabino.miranda@infotec.mx, sabinomiranda@gmail.com (S. Miranda-Jiménez), mario.graff@infotec.mx (M. Graff), dmoctezuma@centrogeo.edu.mx (D. Moctezuma), ranyart@dep.fie.umich.mx (R.R. Suárez), osanchez@centrogeo.edu.mx (O.S. Siordia).

http://dx.doi.org/10.1016/j.patrec.2017.05.024
0167-8655/© 2017 Elsevier B.V. All rights reserved.

Table 1
Parameter list and a brief description of the functionality.

cross-language features		
name	values	description
del-d1	yes, no	If it is enabled then the sequences of repeated symbols are replaced by a single occurrence of the symbol.
del-diac	yes, no	Determines if diacritic symbols, e.g., accent symbols, should be removed from the text.
emo	remove, group, none	Controls how emoticons are handled, i.e. removed, grouped by expressed emotion, or nothing.
num	remove, group, none	Controls how numbers are handled, i.e., removed, grouped into a special tag, or nothing.
url	remove, group, none	Controls how URLs are handled, i.e., removed, grouped into a special tag, or nothing.
usr	remove, group, none	Controls how users are handled, i.e., removed, grouped into a special tag, or nothing.
lc	yes, no	Letters are normalized to be lowercase if it is enabled
language dependent features		
name	values	description
stem	yes, no	Determines if words are stemmed.
neg	yes, no	Determines if negation operators in the text are normalized and directly connected with the next content word.
sw	remove, group, none	Controls how *stopwords* are handled, i.e., removed, grouped, or left untouched.
tokenizers		
tokenizer	$\mathcal{P}$(n-words ∪ q-grams)	One item among the power set (discarding the emptyset) of the union of *n-words and *q-grams.
*n-words	{1, 2}	The number of words used to describe a token.
*q-grams	{1, 2, 3, 4, 5, 6, 7}	The length in characters of a token.

languages tested. Finally, even when our method is almost cross-language, it can be extended to take advantage of language dependencies; we also provide experimental evidence of the advantages of using these language-dependent techniques.

The rest of the manuscript is organized as follows. Section 2 describes our proposed Sentiment Analysis method. Section 3 describes the datasets and contests used to test our approach; whereas, the experimental results, and, the discussion are presented on Section 4. Finally, the conclusions are presented in Section 5.

2. Our approach: multilingual polarity classification

We propose a method for multilingual polarity classification that can serve as a baseline as well as a framework to build more complex sentiment analysis systems due to its simplicity and availability as an open source software.[1] This baseline algorithm for multilingual Sentiment Analysis (B4MSA) was designed with the purpose of being multilingual and easy to implement. Nonetheless, B4MSA is not a naïve baseline as shown by the results obtained on several international competitions.

In a nutshell, B4MSA starts by applying text-transformations to the messages, then transformed text is represented in a vector space model (see Subsection 2.4), and finally, a Support Vector Machine (with a linear kernel) is used as the classifier. B4MSA uses a number of text transformations that are categorized in cross-language features (see Subsection 2.1), language dependent features (see Subsection 2.2) and tokenizers (see Subsection 2.3). It is important to note that, all the text-transformations considered are either simple to implement or there is a well-known library (e.g.[9,23]) to use them. Furthermore, in order to maintain the cross-language property, we limit ourselves to not use additional knowledge, this includes knowledge from affective lexicons or models based on distributional semantics.

To obtain the best performance, one needs to select those text-transformations that work best for a particular dataset, therefore, B4MSA uses a simple random search and hill-climbing (see Subsection 2.5) in the space of text-transformations to free the user from this delicate and time-consuming task. Table 1 gives a summary of the text-transformations used as well as their parameters associated. We consider seven common text transformations for all languages (cross-language features); three particular text transformations that depend on the specific language (language dependent features); and two tokenizers that denote how texts are split after applying the cross-language and dependent language features.

2.1. Cross-language features

We defined cross-language features as a set of features that could be applied to the majority of languages, not only related language families such as Germanic languages (English, German, etc.), or Romance languages (Spanish, Italian, etc.), among others; this is done by using features such as punctuation, diacritics, symbol duplication, case sensitivity, etc. Later, the combination of these features will be explored to find the best configuration for a given classifier.

2.1.1. Spelling features

Generally, Twitter messages are full of slang, misspelling, typographical and grammatical errors among others; in order to tackle these aspects we consider different parameters to study this effect. The following transformations are ones considered as spelling features. *Punctuation* (*del-punc*) considers the use of symbols such as question mark, period, exclamation point, commas, among other spelling marks. *Diacritic symbols* (*del-diac*) are commonly used in languages such as Spanish, Italian, Russian, etc., and its wrong usage is one of the main sources of orthographic errors in informal texts; this parameter considers the use or absence of diacritical marks. *Symbol reduction* (*del-d1*), usually, Twitter messages use repeated characters to emphasize parts of the word to attract user's attention. This aspect makes the vocabulary explodes. The strategy used is to replace the repeated symbols by one occurrence of the symbol. *Case sensitivity* (*lc*) considers letters to be normalized in lowercase or to keep the original source.

2.1.2. Emoticon (emo) feature

We classified around 500 most popular emoticons, included text emoticons, and the whole set of unicode emoticons (around 1,600) defined by Unicode [27] into three classes: positive, negative and neutral. Each emoticon is grouped under its corresponding polarity word defined by the class name.

Table 2 shows an excerpt of the dictionary that maps emoticons to their corresponding polarity class.

2.2. Language dependent features

The following features are language dependent because they use specific information from the language concerned. Usually, the

[1] https://github.com/INGEOTEC/b4msa.

Download English Version:

https://daneshyari.com/en/article/4970059

Download Persian Version:

https://daneshyari.com/article/4970059

Daneshyari.com