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.

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.

Table 2
An excerpt of the mapping table from Emoticons to its polarity words.

:)	:D	:P	→	pos
:(	:-(	:’(	→	neg
:-|	U_U	-.-	→	neu

use of stopwords, stemming and negations are traditionally used in Sentiment Analysis. The users of this approach could add other features such as part of speech, affective lexicons, etc. to improve the performance [16].

2.2.1. Stopwords (sw) feature

In many languages, there is a set of extremely common words such as determiners or conjunctions (e.g. *the* or *and*) which help to build sentences but do not provide any meaning for themselves. These words are known as *Stopwords*, and they are removed from text before any attempt to classify them. Generally, a stopword list is built using the most frequent terms taken from a huge document collection. We used the Spanish, English and Italian stopword lists included in the NLTK Python package [9] in order to identify them.

2.2.2. Stemming (stem) feature

Stemming is a well-known heuristic process in Information Retrieval field that chops off the end of words, and, often, includes the removal of derivational affixes. This technique uses the morphology of the language coded in a set of rules. These rules are applied to find out word stems, the effect is to reduce the vocabulary by collapsing derivationally related words. In our study, we use the Snowball Stemmer for Spanish and Italian, and the Porter Stemmer for English; all of them are implemented in NLTK package [9].

2.2.3. Negation (neg) feature

Negation markers might change the polarity of the message. Thus, we attached the negation clue to the nearest word, similar to the approaches used in [26]. A set of rules was designed for common negation structures that involve negation markers for Spanish, English and Italian.[2] The rules (regular expressions) are processed in order, and their purpose is to negate the nearest word to the negation marker using only the information on the text, e.g., avoiding mainly pronouns and articles. For example, in the sentence *El coche no es bonito* (The car is not nice), the negation marker *no* and *not* (for English) is attached to its adjective *no_bonito* (*not_nice*).

2.3. Tokenizers

The last step, before creating the vector space model, is to split the text into chunks in a range of lengths, this process is known as tokenize. Tokenizers are selected based on word-based $n-$grams and character-based $q-$grams, in any combination.

2.3.1. Word-based n-grams (n-words) feature

Word based n-grams, or simply n-words, are word sequences widely used in many NLP tasks, as well as Sentiment Analysis (see [26] and [11]). To compute the n-words, the text is tokenized, and word n-grams are calculated from tokens. For example, let $T=$`‘‘the lights and shadows of your future’’` be the text, so its 1-words (unigrams) are each word alone, and its 2-words (bigrams) set are the sequences of two words (namely W_2^T), and so on. For example, given T then set W_2^T is {`the lights, lights and, and shadows, shadows of, of your, your future`}. In general, given a text with m words, it is obtained a set containing at most $m-n+1$ elements. Generally, n-words are used up to 2 or 3-words because it is uncommon to find, between texts, good matches of word sequences greater than 3 or 4 words [14]. Among our features, we allow unigrams (1-words) and bigrams (2-words).

2.3.2. Character-based q-grams (q-grams)

In addition to the traditional n-words representation, we represent the resulting text as character q-grams, or simply q-grams. A q-grams is an agnostic language transformation that consists in representing a document by all its substring of length q. For example, let $T=$ “`abra_cadabra`″ be the text, its 3-grams set is

$$Q_3^T = \{\texttt{abr, bra, ra\_, a\_c, \_ca, aca, cad, ada, dab}\},$$

so, given a text with m characters, it is obtained a set with at most $m-q+1$ elements. Notice that this transformation handles whitespaces as part of the text. Since there will be q-grams connecting words, in some sense, applying q-grams to the entire text can capture part of the syntactic and contextual information in the sentence. The rationale of q-grams is also to tackle misspelled sentences from the approximate pattern matching perspective [21]. We allow $q-$grams of length 1–7 characters.

2.4. Text representation

After text-transformations, the text needs to be represented in suitable form in order to use a traditional classifier such as SVM. It was decided to select the well known vector representation of a text given its simplicity and powerful representation. Particularly, the representation used is Frequency-Inverse Document Frequency (TF-IDF), which is a well-known weighting scheme in NLP. TF-IDF computes a weight that represents the importance of tokens inside a document in a collection of documents, i.e., how frequently these appear across multiple documents. In a TF-IDF scheme common words such as *the* and *in*, which appear in many documents, will have a low score, and words that appear frequently in a single document will have high score. This weighting scheme selects the terms that represent a document.

2.5. Parameter optimization

The model selection, sometimes called hyper-parameter optimization, is essential to ensure the performance of a sentiment classifier. In particular, our approach is highly parametric; in fact, we use such property to adapt to several languages. Table 1 summarizes the parameters and their valid values. The search space contains more than 331 thousand configurations when limited to multilingual and language independent parameters; while the search space reaches close to 4 million configurations when we add our three language-dependent parameters. Depending on the size of the training set, each configuration needs several minutes on a commodity server to be evaluated; thus, an exhaustive exploration of the parameter space can be quite expensive making the approach useless in practice. To tackle the efficiency problems, we perform the model selection using two hyper-parameter optimization algorithms.

The first corresponds to *Random Search*, described in depth in [8]. Random search consists on randomly sampling the parameter space and select the best configuration among the sample. The second algorithm consists on a *Hill Climbing* [7,10] implemented with a memory to avoid testing a configuration twice. Algorithm 1 shows our *H+M* approach that consists of taking a pivoting configuration with random search (lines 2–4), explore the configuration’s

[2] For instance, negation markers used for Spanish are *no* (not), *nunca, jamás* (never), and *sin* (without).

Table 3
Details of datasets for each competition tested in this work.

language	dataset	positive	neutral	negative	none	total
SemEval'15	Training	2800	3661	1060	–	7,521
(English)	Development	446	580	262	–	1288
	Gold	841	824	298	–	1,963
SemEval'16	Training	3094	2043	863	–	6000
(English)	Development	844	765	391	–	2000
	Gold	7059	10,342	3231	–	20,632
TASS'15	Training	2884	670	2,182	1,482	7218
(Spanish)	Development	–	–	–	–	-
	Gold 1K	363	22	268	347	1,000
	Gold 60K	22,233	1305	15,844	21,416	60,798
SENTIPOL'14	Training	969	320	1671	1541	4,501
(Spanish)	Development	–	–	–	–	–
	Gold	453	113	754	607	1,927
Arabic [18,25]	Unique	448	202	1350	–	2,000
German [19]	Unique	23,860	50,368	17,274	–	91,502
Portuguese [19]	Unique	24,595	29,357	32,110	–	86,062
Russian [19]	Unique	19,238	28,665	21,197	–	69,100
Swedish [19]	Unique	13,265	15,410	20,580	–	49,255

Algorithm 1 Searching for models in the parameter space with the H+M approach.

```
Input: configuration space $\mathcal{X}$, the size of the random-search's sampling $ss$
Output: the selected configuration $c$

1:  Let $M$ be a hash table that stores the evaluated models and its related score
2:  Let $\mathcal{C} \subset \mathcal{X}$, $ss = |\mathcal{C}|$
3:  Initialize $c$ as $\max \arg_{u \in \mathcal{C}} \mathsf{score}(u)$
4:  Add all configurations and scores of $\mathcal{C}$ into $M$
5:  repeat
6:    Let $prev = M[c]$
7:    for all $u \in \mathsf{neighborhood}(c)$ do
8:      if $u \notin M$ then
9:        $M[u] \leftarrow \mathsf{score}(u)$
10:       if $M[c] < M[u]$ then
11:         $c \leftarrow u$
12:       end if
13:     end if
14:   end for
15: until $prev = M[c]$
16: return $c$
```

neighborhood (loop starting at line 7), and greedily moves to the best neighbor (lines 10–12) under a score function. The process is repeated until no improvement is possible.

The configuration neighborhood $\mathsf{neighborhood}(\cdot)$ is defined as the set of configurations such that these differ in just one parameter's value, see Table 1. This rule is strengthened for *tokenizer* to differ in a single internal value not in the whole parameter value. More precisely, let t be a valid *tokenizer* and $\mathsf{neighborhood}(t)$ the set of valid values for neighborhoods of t,[3] then $|t \cup s| \in \{|t|, |t| + 1\}$ and $|t \cap s| \in \{|t|, |t| - 1\}$ for any $s \in \mathsf{neighborhood}(t)$. For example, let $T = \{1, 2, 3, 4\}$ be the set of tokenizers, e.g. the numbers can be the sizes of the q-grams, then $\mathsf{neighborhood}(\{1, 2, 3\}) = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3, 4\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$.

To guarantee a better or equal performance than random search, the *H+M* process starts with the best configuration found in the random search (lines 2–4). As a rule of thumb, using $ss = 32$, or 64, improve the performance of random search in most cases (see Section 4). Nonetheless, this simplification and performance boosting come along with possible higher optimization times. Finally, the performance of each configuration is obtained using a cross-validation technique on the training data, embedded into the score function. The performance is measured with metrics used in classification such as: accuracy, score F_1, and recall, among others.

3. Datasets and contests

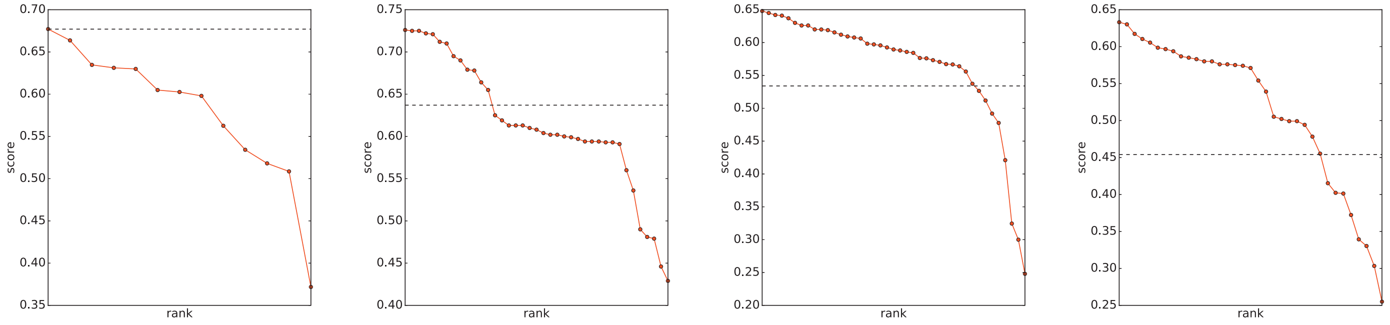
Nowadays, there are several international competitions related to text mining, which includes diverse tasks such as polarity classification (at different levels), subjectivity classification, entity detection, and irony detection, among others. These competitions are relevant to measure the potential of different proposed techniques. In this case, we focused on polarity classification task, hence, we developed a baseline method with an acceptable performance achieved in three different contests, namely, TASS'15 (Spanish) [28], SemEval'15-16 (English) [20,24], and SENTIPOLC'14 (Italian) [6]. Besides, our approach was tested with other languages (Arabic, German, Portuguese, Russian, and Swedish) to show that it is feasible to use our framework as basis for building more complex sentiment analysis systems. The datasets and results from the rest of the languages can be seen in [18,19,25].

Table 3 presents the details of each of the competitions considered as well as the other languages tested. It can be observed, from the table, the number of examples as well as the number of instances for each polarity level, namely, positive, neutral, negative and none. The training and development (only in SemEval) sets are used to train the sentiment classifier, and the gold set is used to test the classifier. Arabic, German, Portuguese, Russian, and Swedish datasets were tested using a cross-validation (10 folds) to be able to compare with the reported literature. The performance of the classifier is presented using different metrics depending on the competition. SemEval uses the average of score F_1 of positive and negative labels, TASS uses the accuracy and SENTIPOLC uses a custom metric (see [6,20,24,28]).

4. Experimental results

We tested our framework on two kinds of datasets. On the one hand, we compare our performance on three languages having well-known sentiment analysis contests; here, we compare our work against competitors of those challenges. On the other hand, we selected five languages without popular opinion mining contests; for these languages, we compare our approach with research works reporting the used corpus.

[3] Notice the notation abuse to define the neighborhood function for tokenizers instead of full configurations.



(a) SENTIPOLC’14

(b) TASS’15

(c) SemEval’15

(d) SemEval’16

name	language	classes	challenge’s score	rank	score	acc.	macro F_1	$\left(F_1^{\text{pos}} + F_1^{\text{neg}}\right)/2$
SENTIPOLC’14	italian	{pos, neg, none, mix}	custom (see [6])	2 / 14	0.677	0.610	0.483	0.630
TASS’15	spanish	{pos, neg, neu, none}	accuracy	14 / 40	0.637	0.637	0.498	0.697
SemEval’15	english	{pos, neg, neu}	$\left(F_1^{\text{pos}} + F_1^{\text{neg}}\right)/2$	34 / 42	0.534	0.629	0.584	0.534
SemEval’16	english	{pos, neg, neu}	$\left(F_1^{\text{pos}} + F_1^{\text{neg}}\right)/2$	28 / 36	0.454	0.534	0.477	0.454

(e) B4MSA’s performance summary; the last columns correspond to three standard scores.

Fig. 1. The performance listing in four difference challenges. The horizontal lines appearing in a) to d) correspond to B4MSA’s performance. All scores values were computed using the official gold-standard and the proper score function for each challenge.

4.1. Performance on sentiment analysis contests

Fig. 1 shows the performance on four contests, corresponding to three different languages. The performance corresponds to the multilingual set of features, i.e., we do not use language-dependent techniques.

Fig. 1(a)–(d) illustrates the results on each challenge, all competitors are ordered in score’s descending order (higher is better). The achieved performance of our approach is marked with a horizontal line on each figure. Fig. 1(e) briefly describes each challenge and summarizes our performance on each contest; also, we added three standard measures to simplify the analysis to the reader.

The winner method in SENTIPOLC’14 (Italian) is reported in [5]. This method uses three groups of features: keyword and micro-blogging characteristics, Sentiment Lexicons, SentiWordNet and MultiWordNet, and Distributional Semantic Model (DSM) with a SVM classifier. In contrast with our method, in [5] three external sentiment lexicons dictionaries were employed, i.e., external information.

In TASS’15 (Spanish) competition, the winner reported method was [17], which proposed an adaptation based on a tokenizer of tweets *Tweetmotif* [15], Freeling [22] as lemmatizer, entity detector, morphosyntactic labeler and a translation of the Afinn dictionary. In contrast with our method, [17] employs several complex and expensive tools. In this task, we reached the fourteenth position with an accuracy of 0.637. Fig. 1(b) depicts that B4MSA’s performance is over two-thirds of the competitors.

The remaining two contests correspond to the SemEval’15-16. The B4MSA performance in SemEval is depicted in Fig. 1(c) and (d); here, B4MSA does not perform as well as in other challenges, mainly because, contrary to other challenges, SemEval promotes the enrichment of the official training set. Nonetheless, in order to be consistent with the rest of the experiments, B4MSA uses only the official training set. The results can be significantly improved using larger training sets; for example, joining SemEval’13 and SemEval’16 training sets, we can reach 0.54 for SemEval’16, which improves the B4MSA’s performance (see Table 1).

In SemEval’15, the winner method is [12], which combines three approaches among the participants of SemEval’13, teams: NRC-Canada, GU-MLT-LT and KLUE, and from SemEval’14 the participant TeamX all of them employing external information. In SemEval’16, the winner method (see [13]) was composed with an ensemble of two subsystems both based on convolutional neural networks. The first subsystem was created using 290 million tweets, and the second one was fed with 150 million tweets. All these tweets were selected from a very large unlabeled dataset through distant supervision techniques.

Table 4 shows the multilingual set of techniques and the set with language-dependent techniques. For each, we optimized the set of parameters through *Random Search* and $H+M$ (see Section 2.5). The reached performance is reported using both cross-validation and the official gold-standard. Please notice how $H+M$ consistently reaches better performances, even on small sampling sizes. The sampling size is indicated with subscripts in Table 4. Note that, in SemEval challenges, the cross-validation performances are higher than those reached by evaluating the gold-standard, mainly because the gold-standard does not follow the distribution of training set. This can be understood because the rules of SemEval promote the use of external knowledge.

Table 5 compares our performance on five different languages; we do not apply language-dependent techniques. For each comparison, we took a labeled corpus from [25] (Arabic) and [19] (the remaining languages). According to author’s reports, all tweets were manually labeled by native speakers as pos, neg, or neu. The Arabic dataset contains 2000 items; the other datasets contain from 58 thousand tweets to more than 157 thousand tweets. We were able to fetch a fraction of the original datasets; so, we drop the necessary items to hold the original class-population ratio. The ratio of tweets in our training dataset, respect to the original dataset, is indicated beside the name. As before, we evaluate our algorithms through a 10-fold cross validation.

In [18,25], the authors study the effect of translation in sentiment classifiers; they found better to use native Arabic speakers as annotators than fine-tuned translators plus fine-tuned English sentiment classifiers. In [19], the idea is to measure the effect of the agreement among annotators on the production of a sentiment-analysis corpus. On the technical side, both papers use fine tuned classifiers plus a variety of pre-processing techniques to prove their claims. Table 5 supports the idea of choosing B4MSA as a bootstrapping sentiment classifier because, overall, B4MSA

Table 4
B4MSA's performance on cross-validation and gold standard. The subscript at right of each score stands for the random-search's parameter (sampling size) needed to find that value.

Dataset		Multilingual Parameters				Language-Dependent Parameters			
		Random Search		H+M Search		Random Search		H+M Search	
		cross-val.	gold-std.	cross-val.	gold-std.	cross-val.	gold-std.	cross-val.	gold-std.
SENTIPOLC	'14	–	0.678_{256}	–	0.677_{16}	–	0.675_{8}	–	0.674_{256}
TASS	'15	0.643_{128}	0.636_{128}	0.648_{8}	0.637_{8}	0.644_{256}	0.635_{256}	0.649_{32}	0.637_{32}
SemEval	'15	0.585_{256}	0.530_{256}	0.590_{8}	0.534_{8}	0.590_{256}	0.520_{256}	0.596_{128}	0.528_{128}
SemEval	'16	0.575_{64}	0.456_{64}	0.578_{64}	0.454_{64}	0.580_{256}	0.462_{256}	0.583_{256}	0.462_{256}

Table 5
Performance on multilingual sentiment analysis (not challenges). B4MSA was restricted to use only the multilingual set of parameters.

language		F_1	$(F_1^{pos} + F_1^{neg})/2$	acc
Arabic	Salameh et al. [25]	–	–	0.787
	Saif et al. [18]	–	–	0.794
	B4MSA (100%)	0.642	0.781	0.799
German	Mozetič et al. [19]	–	0.536	0.610
	B4MSA (89%)	0.621	0.559	0.668
Portuguese	Mozetič et al. [19]	–	0.553	0.507
	B4MSA (58%)	0.550	0.591	0.555
Russian	Mozetič et al. [19]	–	0.615	0.603
	B4MSA (69%)	0.754	0.768	0.750
Swedish	Mozetič et al. [19]	–	0.657	0.616
	B4MSA (93%)	0.680	0.717	0.691

reaches superior performances regardless of the language. Our approach achieves those performance's levels since it optimizes a set of parameters carefully selected to work on a variety of languages and robust to informal writing. The latter problem is not properly tackled in many cases.

4.2. Feature analysis

Table 6 shows the empirical probability of using some particular feature among the best ten configurations. The table shows this structural analysis for SENTIPOLC'14, TASS'15, SemEval'15 and SemEval'16. Since, we used language dependent features to solve these benchmarks, then we have taken into account the possibility of selecting *sw, neg,* and *stem* features. It can be observed from the table that removal of stopwords is never selected, the negation is popular for TASS'15 and SemEval'15, and not so much for the rest. The use of stemming is recommended for SemEval'15, but it is not for the others. Note that SemEval'15 and SemEval'16, both in English, are quite different in its language features. Other features are not very different for all benchmarks. It is interesting to note that *emo* feature is almost never used, in some way, the effect of this feature is replaced by other features, like q-grams of size 1 or 2, which are the typical lengths of emoticons.

Tokenizers are among the features with more variation; then it is interesting to focus on them. Almost all of them use 1-words (unigrams), and many of them use 2-words (bigrams). There is no a favorite tokenizer, but most tokenizers are used in top-10 sentiment classifiers. As commented, small q-grams can be used to replace the *emo* feature while the rest can be capturing word connections and full words.

Table 7 shows the composition of best ten configurations for Arabic, German, Portuguese, Russian, and Swedish benchmarks; for these datasets, we do not use language dependent features. As before, the most varying features are the tokenizers. Here, we can see that 2-words and 1-words are less used than before. Also, notice that Arabic benchmark does not use them at all; the same happens for the Russian dataset. In contrast to Table 6, the large q-grams are not popular in Table 7, yet the smaller ones are almost always used (size of 1 to 3). This can be a reflect of the dataset composition, or simply the effect of removing the language dependent functionalities.

It is worth to know that our parameter optimization strategy will try to adapt to the training set, no matter the underlying language. Whenever we allow the use of language dependent features, this can be masked by the fact that a language feature is used. However, it will always try to improve the score function without regarding the particular procedure to do it. In some sense, this language dissociation is also supported by the SVM classifier, since it works through a kernel function and never regards on individual features to work.

It is possible that a fine selection of the available features, along with a more transparent classifier, can produce models that could help to improve the understanding of the sentiment distribution of a particular dataset, and ultimately, of a language. However, that study is left as an open issue, since it is beyond the scope of this contribution.

5. Conclusions

We presented a simple to implement multilingual framework for polarity classification whose main contributions are in two aspects. On the one hand, our approach can serve as a baseline to compare other classification systems. It considers techniques for text representation such as spelling features, emoticons, word-based n-grams, character-based q-grams and language dependent features. On the other hand, our approach is a framework for practitioners or researchers looking for a bootstrapping sentiment classifier method to build more elaborated systems.

Besides the text-transformations, the proposed framework uses a SVM classifier (with a linear kernel), and, hyper-parameter optimization using random search and *H+M* over the space of text-transformations. The experimental results show good overall per-

Table 6
Empirical probability of using a particular feature and tokenizer in the best 10 configurations of each competition, see Fig. 1(e). The configurations were evaluated by the H+M algorithm during the optimization process.

name	del-d1	del-diac	emo	lc	num	url	usr	sw	neg	stem	n = 2	n = 1	q = 1	q = 2	q = 3	q = 4	q = 5	q = 6	q = 7
SENTIPOLC'14	0.6	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.2	0.0	0.2	1.0	0.8	1.0	1.0	0.2	0.8	0.4	0.6
TASS'15	0.4	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.4	1.0	1.0	0.0	1.0	1.0	0.0
SemEval'15	1.0	0.9	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.4	1.0	1.0	0.0	1.0	1.0	0.4	0.4	1.0
SemEval'16	0.8	0.6	0.0	1.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.8	1.0	0.0	0.0	1.0	0.8	0.8	1.0

Table 7
Empirical probability of using a particular feature and tokenizer in the best 10 configurations of the multilingual datasets listed in Table 5. The configurations were evaluated by the H+M algorithm during the optimization process.

name	del-d1	del-diac	emo	lc	num	url	usr	n = 2	n = 1	q = 1	q = 2	q = 3	q = 4	q = 5	q = 6	q = 7
Arabic	0.2	0.8	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.9	0.0	0.0	0.0
German	0.3	1.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0
Portuguese	0.9	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.9	0.9	1.0	0.1	0.0	0.0	0.0
Russian	0.8	0.8	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	1.0	0.0
Swedish	0.2	0.0	0.0	1.0	0.0	0.0	0.0	0.6	1.0	1.0	1.0	0.9	1.0	0.0	0.0	1.0

formance in all international contests considered, and the best results in the other five languages tested.

It is important to note that all the methods that outperformed B4MSA in the sentiment analysis contests use extra knowledge (lexicons included) meanwhile B4MSA uses only the information provided by each contest. In future work, we will extend our methodology to include extra-knowledge to improve the performance.

Acknowledgments

We would like to thank Valerio Basile, Julio Villena-Roman, and Preslav Nakov for kindly give us access to the gold-standards of SENTIPOLC’14, TASS’15 and SemEval 2015 & 2016, respectively. The authors also thank Elio Villaseñor for the helpful discussions in early stages of this research.

References

[1] M. Araujo, J. Reis, A. Pereira, F. Benevenuto, An evaluation of machine translation for multilingual sentence-level sentiment analysis, in: Proceedings of the 31st Annual ACM Symposium on Applied Computing, in: SAC’16, ACM, New York, NY, USA, 2016, pp. 1140–1145.

[2] D. Bal, M. Bal, A. van Bunningen, A. Hogenboom, F. Hogenboom, F. Frasincar, Sentiment Analysis with a Multilingual Pipeline, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 129–142.

[3] A. Balahur, M. Turchi, Multilingual sentiment analysis using machine translation? in: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, in: WASSA’12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 52–60.

[4] A. Balahur, M. Turchi, Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis, Comput. Speech Lang. 28 (1) (2014) 56–75.

[5] P. Basile, N. Novielli, Uniba at evalita 2014-sentipolc task: predicting tweet sentiment polarity combining micro-blogging, lexicon and semantic features, in: Proceedings of the 4th Evaluation campaign of Natural Language Processing and Speech Tools for Italian (EVALITA’14), 2014. Pisa, Italy.

[6] V. Basile, A. Bolioli, M. Nissim, V. Patti, P. Rosso, Overview of the Evalita 2014 SENTIment POLarity classification task, in: Proceedings of the 4th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA’14), 2014. Pisa, Italy.

[7] R. Battiti, M. Brunato, F. Mascia, Reactive Search and Intelligent Optimization, 45, Springer Science & Business Media, 2008.

[8] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[9] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, O’Reilly Media, 2009.

[10] E.K. Burke, G. Kendall, et al., Search Methodologies, Springer, 2005.

[11] Z. Cui, X. Shi, Y. Chen, Sentiment analysis via integrating distributed representations of variable-length word sequence, Neurocomputing 187 (2015) 126–132.

[12] M. Hagen, M. Potthast, M. Bchner, B. Stein, Webis: an ensemble for Twitter sentiment detection, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015, Association for Computational Linguistics, 2015, pp. 582–589.

[13] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, M. Jaggi, Swisscheese at semeval-2016 task 4: sentiment classification using an ensemble of convolutional neural networks with distant supervision, in: Proceedings of the 10th International Workshop on Semantic Evaluation, in: SemEval’16, Association for Computational Linguistics, San Diego, California, 2016.

[14] D. Jurafsky, J.H. Martin, Speech and Language Processing (2nd Edition), Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.

[15] M. Krieger, D. Ahn, Tweetmotif: exploratory search and topic summarization for Twitter, in: In Proc. of AAAI Conference on Weblogs and Social, 2010.

[16] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015. ISBN: 1-107-01789-0. 381 pages.

[17] F.P.L.-F. Hurtado, D. Buscaldi, Elirf-upv en tass 2015: Anlisis de sentimientos en Twitter, in: J. Villena Román, G. Morera, Janine, G. Cumbreras, M. Ángel, M. Cámara, Eugenio, M. Valdivia, M. Teresa, U.n. López, L. Alfonso (Eds.), TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), 2015, pp. 75–79.

[18] S.M. Mohammad, M. Salameh, S. Kiritchenko, How translation alters sentiment, J. Artif. Intell. Res. 55 (2016) 95–130.

[19] I. Mozetič, M. Grčar, J. Smailović, Multilingual Twitter sentiment classification: the role of human annotators, PLoS ONE 11 (5) (2016). E0155036.

[20] P. Nakov, A. Ritter, S. Rosenthal, V. Stoyanov, F. Sebastiani, SemEval-2016 task 4: sentiment analysis in Twitter, in: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval’16, Association for Computational Linguistics, San Diego, California, 2016.

[21] G. Navarro, M. Raffinot, Flexible Pattern Matching in Strings – Practical On-Line Search Algorithms for Texts and Biological Sequences, Cambridge University Press, 2002. ISBN 0-521-81307-7. 280 pages.

[22] L. Padró, E. Stanilovsky, Freeling 3.0: Towards wider multilinguality, in: Proceedings of the Language Resources and Evaluation Conference (LREC 2012), ELRA, Istanbul, Turkey, 2012.

[23] R. Řehůřek, P. Sojka, Software framework for topic modelling with large corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA, Valletta, Malta, 2010, pp. 45–50. http://is.muni.cz/publication/884893/en.

[24] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, V. Stoyanov, Semeval-2015 task 10: sentiment analysis in Twitter, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado, 2015, pp. 451–463.

[25] M. Salameh, S. Mohammad, S. Kiritchenko, Sentiment after translation: a case-study on arabic social media posts, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 767–777.

[26] G. Sidorov, S. Miranda-Jiménez, F. Viveros-Jiménez, A. Gelbukh, N. Castro-Sánchez, F. Velásquez, I. Díaz-Rangel, S. Suárez-Guerra, A.T. no, J. Gordon, Empirical study of machine learning based approach for opinion mining in tweets, in: Proceedings of the 11th Mexican International Conference on Advances in Artificial Intelligence - Volume Part I, in: MICAI’12, Springer-Verlag, Berlin, Heidelberg, 2013, pp. 1–14.

[27] Unicode, Unicode emoji chart, 2016, http://unicode.org/emoji/charts/full-emoji-list.html, Accessed 20-May-2016.

[28] J.V. Román, J.G. Morera, M.A.G. Cumbreras, E.M. Cámara, M.T.M. Valdivia, L.A.U.n. López, Overview of tass 2015, in: V. Román, Julio, G. Morera, Janine, G. Cumbreras, M. Ángel, M. Cámara, Eugenio, M. Valdivia, M. Teresa, U.n. López, L. Alfonso (Eds.), TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), 2015, pp. 13–21.