Expert Systems With Applications 62 (2016) 243–249

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Short text opinion detection using ensemble of classifiers and semantic indexing

Johannes V. Lochter[a,*], Rafael F. Zanetti[b], Dominik Reller[a], Tiago A. Almeida[a]

[a] Department of Computer Science, Federal University of São Carlos – UFSCar, Sorocaba, 18052-780, Brazil
[b] Department of Computer Sciences, University of Wisconsin-Madison – Madison, WI 53703 USA

ARTICLE INFO

Article history:
Received 15 March 2016
Revised 2 June 2016
Accepted 12 June 2016
Available online 16 June 2016

Keywords:
Sentiment analysis
Text normalization
Semantic indexing
Classification
Machine learning

ABSTRACT

The popularity of social networks has attracted attention of companies. The growing amount of connected users and messages posted per day make these environments fruitful to detect needs, tendencies, opinions, and other interesting information that can feed marketing and sales departments. However, the most social networks impose size limit to messages, which lead users to compact them by using abbreviations, slangs, and symbols. As a consequence, these problems impact the sample representation and degrade the classification performance. In this way, we have proposed an ensemble system to find the best way to combine the state-of-the-art text processing approaches, as text normalization and semantic indexing techniques, with traditional classification methods to automatically detect opinion in short text messages. Our experiments were diligently designed to ensure statistically sound results, which indicate that the proposed system has achieved a performance higher than the individual established classifiers.

1. Introduction

Digital inclusion has allowed an increasing number of Internet users, which recently has been responsible for the most success of social networks. In such applications, users are able to share and read information, and perform many activities. Among shared information, users often post opinions and rate products. According to a press release of ComScore[1], online reviews have a significant impact on purchasing behavior. Consequently, companies noticed how important it is to be able to analyze a huge amount of messages in a fast way to discover tendencies and opinion of users.

The employment of classification methods in opinion detection were presented in some works (Denecke, 2008; Luo, Zeng, & Duan, 2016; Pang, Lee, & Vaithyanathan, 2002). However, in most cases, it is still very difficult to identify the polarity of text samples extracted from social networks because, besides being very short, they are often rife with idioms, slang, symbols, emoticons and abbreviations which make even tokenization a challenge task (Denecke, 2008).

Noise in text messages can appear in different ways. The following phrase offers an example: *"dz ne1 knw h2 ripair dis terrible LPT? :("*. There are misspelled words *"dz,ne1,knw,h2,dis"*, abbreviation *"LPT"* and symbol *":("*. In order to transcribe such phrase to a proper English grammar, a *Lingo* dictionary[2] would be needed along with a standard dictionary, which associates each slang, symbol or abbreviation to a correct term. After a step of text normalization, the input phrase would be translated to *"Does anyone know how to repair this terrible printer? :("* and the symbol at the end would mean the author has a sad or dissatisfied sentiment about the product.

In addition to noisy messages, there are other well-known problems described in literature such as sarcasm, ambiguous words in context (polysemy) and different words with the same meaning (synonymy). When such cases are properly handled, better results can be achieved (Mostafa, 2013; Pang & Lee, 2008).

Both synonymy and polysemy problems can have their effect minimized by semantic indexing for word sense disambiguation (Navigli & Ponzetto, 2012; Taieb, Aouicha, & Hamadou, 2013). Such dictionaries associate meanings to words by finding similar terms given the context of message. In general, the effectiveness of applying such dictionaries relies in the quality of terms extracted from samples. However, common tools for natural language pro-

* Corresponding author.
E-mail addresses: jlochter@acm.org, alemaoyo@gmail.com (J.V. Lochter), ferrazzanett@wisc.edu (R.F. Zanetti), dreller@acm.org (D. Reller), talmeida@ufscar.br (T.A. Almeida).

[1] ComScore press release. Available at http://goo.gl/PRIHmS, accessed in March 30, 2015.

[2] Lingo is an abbreviated language commonly used on Internet applications, such as chats, emails, blogs and social networks.

http://dx.doi.org/10.1016/j.eswa.2016.06.025
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

cessing can not be suitable to deal with short texts, demanding proper tools for working in this context (Bontcheva et al., 2013; Maynard, Bontcheva, & Rout, 2012).

Even after dealing with problems of polysemy and synonymy, resulting terms may not be enough to detect opinion because the original messages are usually very short. Some recent works recommend to employ ontology models to analyze each term and find associated new terms (with the same meaning) to enrich original sample with more features (Kontopoulos, Berberidis, Dergiades, & Bassiliades, 2013).

Terms achieved by ontology models and semantic indexing (called expansion process) are more representative for classification methods if they can be related to an individual polarity. This way, recent works also demonstrate that lexical dictionaries can enhance classification performances (Mostafa, 2013; Nastase & Strube, 2013).

Original samples can be processed by different text processing techniques and resulting text samples become inputs to classification methods. Since there are several techniques to perform feature processing and different established classification methods, an ensemble system that naturally integrates these approaches could overcome individual drawbacks, achieve better hypothesis and consequently enhance the overall prediction performance. Ensemble strategies are commonly applied in literature to combine outputs of several classifiers in an integrated final output (Dietterich, 2000; Wang, Sun, Ma, Xu, & Gu, 2014; Xia, Zong, & Li, 2011).

In this scenario, we have designed and evaluated an ensemble system to perform opinion detection in short text messages extracted from social networks. Our model combines text normalization methods along with state-of-the-art natural language processing techniques to improve quality of extracted features which are then used by established machine learning approaches. The results demonstrate that our proposal clearly outperforms established methods available in literature.

This paper is organized as follows: Section 2 presents the most relevant related work. Text normalization and semantic indexing techniques are described in Section 3. Section 4 presents the proposed ensemble system. Experimental methodology is described in Section 5. Section 6 presents the achieved results and main conclusions are provided in Section 7.

2. Related work

Opinion detection is the task of analyzing huge amounts of information from thousands (or millions) of users to detect the majority opinion about anything in discussion. The understanding and fast reaction about such opinions allows companies to guide their marketing and to aid in decision making (Mostafa, 2013; Pang et al., 2002). According to results available in literature, this task is far from being properly solved due to many reasons, such as difficulties to deal with sarcasm, irony, and sentences with multiple polarities. In addition, another important well-known problem is related to the amount and quality of features extracted from messages. Often, text messages extracted from social networks are short and usually rife with noise (slangs, symbols, abbreviations, and so on), causing bad vector representation that decreases the classifiers performances (Go, Bhayani, & Huang, 2009; Navigli & Lapata, 2010).

In text categorization, a challenge that remains in dealing with short text is the lack of information about its content. The limit size usually imposed by the channel (e.g., Twitter), not rarely, leads classification methods to face problems like polysemy and synonymy. Polysemy is the capacity for a single term has multiple meanings represented by only one attribute in a feature vector. Synonymy is related to the capacity for multiple terms have same meaning represented by more than one attribute in a feature vector. In this scenario, there are recent works that successfully applied semantic indexing and lexical normalization to avoid these problems in order to improve the quality of features (Nastase & Strube, 2013; Navigli & Ponzetto, 2012).

Lexical normalization or text normalization is the task of replacing lexical variants of standard words and expressions normally obfuscated in noisy texts to their canonical forms, in order to allow further processing of text processing tasks. It is closely related to spell checking, and in fact, many approaches in literature share techniques from this task (Cook & Stevenson, 2009; Xue, Yin, Davison, & Davison, 2011).

Semantic indexing or Query Expansion is the task of replacing words in texts by their synonyms according to the concept the target word belongs to (Hidalgo, Rodríguez, & Pérez, 2005). As an example, the semantic network WordNet represents synonyms sets as following: {car, auto, automobile, machine, motorcar} (a motor vehicle with four wheels) or {car, railcar, railway car, railroad car} (a wheeled vehicle adapted to the rails of railroad) for the word “car”.

Output samples produced by semantic indexing add complexity in the task of identifying the most appropriate concepts for each word in the message given its context. This problem can be handled using *Word Sense Disambiguation* (WSD) which is a popular technique used in deep natural language processing (Agirre & Edmonds, 2006). In this work, we have used the BabelNet semantic network along with WSD unsupervised algorithm (Navigli & Lapata, 2010), following the Semantic Expansion method described in Gómez Hidalgo et al. (2005).

After lexical normalization and semantic indexing, the original noisy samples are processed and expanded by adding new concepts related to the context of terms in sample. Therefore, besides the sample being normalized, it is also enriched with more information in order to aid classification methods to improve their prediction capacities (Kontopoulos et al., 2013; Nastase & Strube, 2013).

The abundance of text processing techniques and classification methods to handle short text messages demands some way to combine them in order to acquire a generic and good hypothesis. In this scenario, an ensemble system is highly recommended to find out a good classification model in an automatic way (Dietterich, 2000).

Ensemble of classifiers is a technique developed to achieve generic hypotheses by combining different classifiers. The ensemble works like a committee in which each classifier is a voting member and the committee produces a final prediction based in their votes. This technique is commonly applied to minimize specific drawbacks, such as overfitting and the curse of dimensionality (Dietterich, 2000). Although ensemble systems can adopt different strategies, they usually achieve better results than individual classifiers (Wang et al., 2014).

Widely adopted, weighted ensemble systems are often found in literature. The effectiveness of these techniques relies on assigning an appropriated weight for each vote. Thus, a less-accurate classifier should not have the same or more a significant vote than a more-accurate one (Kim, Kim, Moon, & Ahn, 2011; Xia et al., 2011).

As short text samples can be processed by different text processing techniques, and moreover, there are several established classification methods recommended to opinion detection, a sophisticated ensemble system that combines these approaches can lead to generic hypotheses and consequently achieve good performance.

3. Text processing techniques

In scenarios where messages are short and rife with idioms, symbols and abbreviations, just employing a simple bag of words

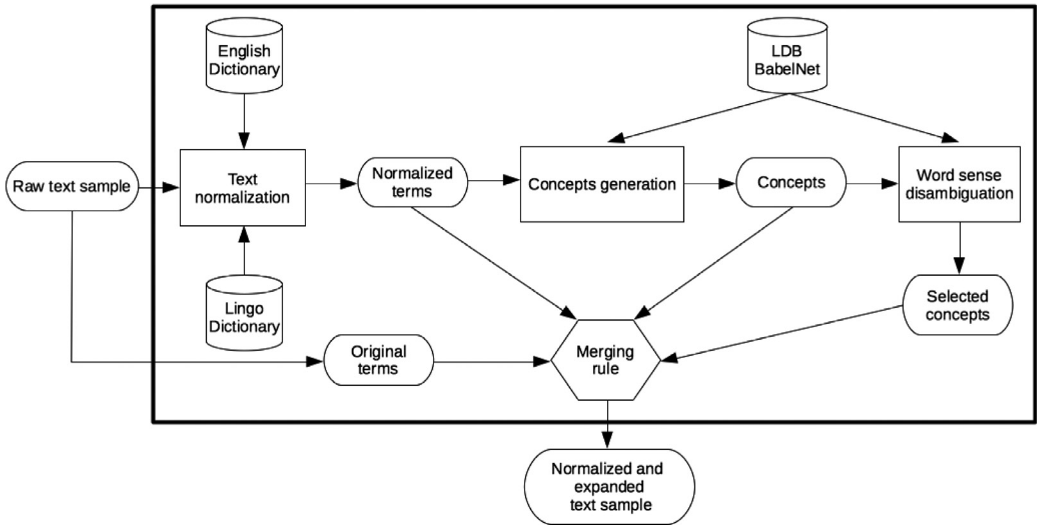

Fig. 1. The original sample is processed by semantic dictionaries and context detection techniques. Each one creates a new normalized or expanded sample. Then, given a merging rule, the samples are joined into a final output represented by a text message with the same semantic content of the original sample.

is not generally enough to achieve satisfactory results (Gabrilovich & Markovitch, 2005). Often, a lexical normalization step is needed to translate obfuscated messages to standard English. Next, as messages can be very short, the amount of features can not be enough to lead to good performance, mainly when problems of synonymy or polysemy are frequent. In this way, semantic dictionaries along with state-of-the-art techniques for context detection are used to expand the original message to one with the same context but being larger in terms of attributes and without ambiguities. For this, we have designed a cascade process[3] in which an input text sample can be processed in three different stages, each one generating a new output representation in turn.

Each stage is briefly described as follows.

3.1. Lexical normalization

In this stage, we have employed two dictionaries to translate words in Lingo, which is the name given to slang and abbreviations commonly used on Internet, to standard English word and phrases. The first dictionary is an English one (e.g., Freeling English dictionary[4]) and it is used to check whether a word is already an English one. If it is found, the word in sample is replaced by its root form. The second dictionary is Lingo (e.g., NoSlang dictionary[5]) itself, which is used to translate a word from Lingo to English.

3.2. Concepts generation

The concepts are provided by the BabelNet repository, which is a modern and huge English semantic dictionary composed by concepts of WordNet and Wikipedia (Navigli & Lapata, 2010). Since it requires English words, *lexical normalization* is applied to certify that each word to be processed is an English one. After that, our method avoids translated words that belong to a common stopwords list (articles and pronouns) to prevent non-representative substitutes and save processing time. If the word is in stopwords list, the original one is kept. Otherwise, it is processed by a semantic dictionary and a list of concepts is computed for such input words.

3.3. Concepts disambiguation

As the amount of concepts computed by each term of message can be large, a disambiguation technique is used to select the most relevant concept according to the context of message. Basically, a word sense disambiguation technique based on the work of Navigli and Ponzetto (2012) was implemented to reach this goal. Then, a merging rule is defined to combine different outputs generated by each text processing step, creating a new final sample that would be used in training and classification stages.

Fig. 1 illustrates the whole process.

3.4. Merging rule

As we have the original sample along with the three above mentioned expansion stages, we have four different parameters to set for defining the merging rule, which are basically answers for the following questions, respectively: 1. *Should it keep the original tokens?*, 2. *Should it perform text normalization?*, 3. *Should it perform the concepts generation?* and 4. *Should it perform the word sense disambiguation?*. As each choice is binary, we have eleven possibilities of settings to expand each sample (including keeping the original samples). Note that, the output of concepts disambiguation process is always a subset of the output produced by concepts generation stage. Table 1 presents each possible set of parameters that can be used in the merging rule.

In our experiments, we have performed all possible merging rules, generating one different expanded dataset for each possible

Table 1
All possible rules that can be used in the expansion method.

Rules	Original terms	Normalized terms	Concepts	Selected concepts
Rule 1	X			
Rule 2	X	X		
Rule 3	X		X	
Rule 4	X		X	X
Rule 5	X	X	X	
Rule 6	X	X	X	X
Rule 7		X		
Rule 8			X	
Rule 9			X	X
Rule 10		X	X	
Rule 11		X	X	X

[3] The proposed text expansion system is publicly available at http://lasid.sor.ufscar.br/expansion/.

[4] *Freeling English dictionary.* Available at: http://devel.cpl.upc.edu/freeling/.

[5] *NoSlang*: Internet Slang Dictionary & Translator. Available at: http://www.noslang.com/dictionary/full/.

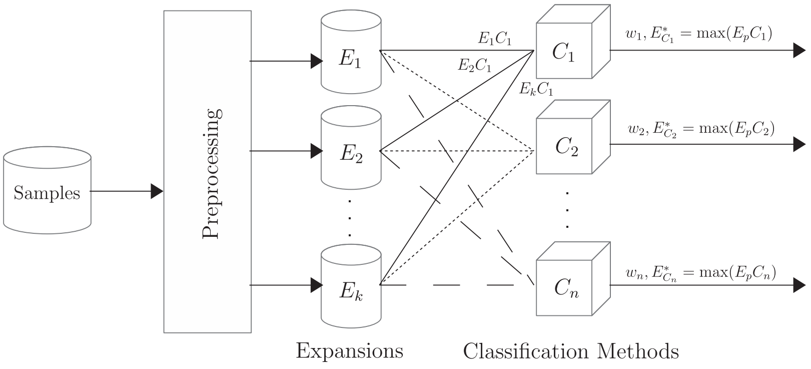

Fig. 2. In model selection, the original dataset is processed by text expansion techniques ($E_1, \ldots, E_k$) which generate expanded datasets. Next, each expanded dataset is evaluated with each classification method ($C_1, \ldots, C_n$) in order to define the best combination $E^*_{c_j} = \max(E_p, C_j)\ \forall\ p \in \{1, \ldots, k\}, j \in \{1, \ldots, n\}$ and w_j corresponds to the weight related to each combination.

Table 2
Example of lexical normalization and semantic expansion using the merging rule [Lexical normalization + concepts disambiguation] which means that outputs of these text processing steps are combined.

Original	plz lemme noe when u get der
Normalization	*please let me know when you get there*
Concepts generation	*please army_of_the_righteous lashkar-e-taiba lashkar-e-tayyiba lashkar-e-toiba let net_ball me knoe knowledge noesis when you get there*
Concepts disambiguation	*please lease me cognition when you get there*
Final sample	*please let lease me know cognition when you get there*

set of parameters. Therefore, the original corpus (Rule 1) and ten created expanded datasets (from Rule 2 to 11) were evaluated.

To provide a brief idea of the whole process, in Table 2 we present an example. Given the original sample "*plz lemme noe when u get der*", the output produced after each step is showed. *Lexical normalization* replaces slangs and abbreviations to their corresponding words in English. While *Concepts generation* obtained all concepts for each word in the original sample, the *Concepts Disambiguation* stage kept only concepts that were semantically relevant to the original sample. Then, defining the merging rule as [Lexical normalization + concepts disambiguation], for instance, we would achieve the final expanded sample "*please let lease me know cognition when you get there*", which we hope will be more suitable for using in machine learning techniques.

4. The ensemble system

The proposed ensemble system is divided in two distinct stages: model selection and classification.

In *model selection*, the first step is to perform a grid search to set the main parameters of methods that compose the system. As this process is time-consuming, only a stratified randomly selected sample set from the original dataset is used. The next step is to employ text processing techniques ($E_1, \ldots, E_k$) to normalize and expand the original input samples. All possible merging rules are used and each one produces a different output. Thus, the resulting expanded datasets are used to train and evaluate each classification method ($C_1, \ldots, C_n$). Next, the performance achieved by each possible combination between expansion steps and classification methods is used to select which merging rule (or expansion step) is more appropriate for each classification method ($E^*_{c_j} = \max(E_p, C_j)\ \forall\ p \in \{1, \ldots, k\}, j \in \{1, \ldots, n\}$). Finally, a weight w_j (degree of confidence) is calculated for each combination j based on its accuracy compared with the best one (Eq. 1). Fig. 2 illustrates the model selection stage.

$$w_j = \frac{1}{\left|\log_2\left(\frac{Acc_j}{max(Acc_j)+T}\right)\right|}, \quad 1 \le j \le n. \tag{1}$$

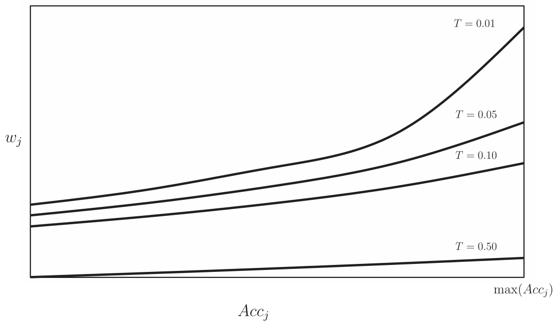

Fig. 3. The constant T impacts the final voting weights between classifiers with different performances. The smaller the value of T, the higher the difference between voting weights of classifiers with low and high accuracies.

In Eq. 1, we use a constant $0 < T \le 1$ to control the balance between high and low weights. The smaller the value of T, the higher the difference between weights of classifiers with low and high accuracies. Therefore, as T approaches zero, the greater the difference between the voting weight w_j of classifier with the highest accuracy ($Acc_j = max(Acc_j)$) and all possible other classifiers with $Acc_j < max(Acc_j)$. On the other hand, as T approaches one, the smaller the difference of voting weights between all classifiers. Fig. 3 illustrates how the value of T impacts the final voting weights of classifiers with different performances in the model selection stage.

After classification methods have been trained and the best model selected, in the *classification stage*, the input sample is processed by prior text processing techniques selected to be employed with each classifier. Then, the expanded and normalized outputs are used as inputs to each classifier that predicts a label with a degree of confidence (voting weight). The final class is then computed by weighted majority vote, as illustrated in Fig. 4.

In resume, we designed an ensemble system able to handle short and/or obfuscated text messages by automatically selecting among many possibilities, the best combination between the state-of-the-art text processing and semantic indexing techniques with a set of established classification methods. With that, the proposed approach is able to create more generic hypothesis and achieve

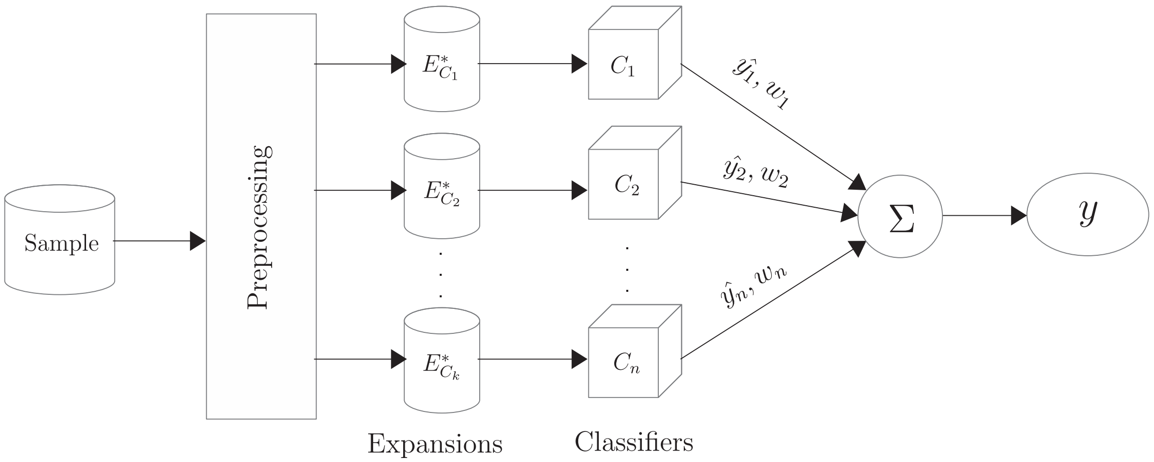

Fig. 4. Once the system learns which text processing techniques (and their combination) (E^*_p) are more suitable for each classification method (C_j), the classifiers are trained (model selection). Then, in the classification stage, given an input sample, it is processed and classified by each model that sends its prediction ($\hat{y}_j$) and its degree of confidence (w_j) to a voting concentrator (Σ). The final prediction is further computed based on its weighted majority vote.

Table 3
Datasets used to evaluate the proposed ensemble system.

Dataset	# Positive	# Negative	Theme
STS-Test (Agarwal, Xie, Vovsha, Rambow, & Passonneau, 2011)	181	177	Misc.
HCR (Speriosu, Sudan, Upadhyay, & Baldridge, 2011)	537	886	Medical
OMD (Shamma, Kennedy, & Churchill, 2009)	709	1195	Politics
SS-Tweet (Thelwall, Buckley, & Paltoglou, 2012)	1252	1037	Misc.
Sanders (Analytics, 2011)	519	572	Misc.
UMICH (UMICH, 2011)	796	669	Movie
IPhone6	371	161	Smartphone
Archeage	724	994	Game
Hobbit	354	168	Movie

Table 4
Classification methods used in our ensemble system and further compared with it.

Classification methods
Bernoulli Naïve Bayes (NB-B) (Almeida, Almeida, & Yamakami, 2011)
Multinomial Naïve Bayes (NB-M) (Almeida et al., 2011)
Gaussian Naïve Bayes (NB-G) (Almeida et al., 2011)
Linear Support Vector Machines (SVM-L) (Cortes & Vapnik, 1995; Haykin, 1998)
Radial Support Vector Machines (SVM-R) (Cortes & Vapnik, 1995; Haykin, 1998)
Polynomial Support Vector Machines (SVM-P) (Cortes & Vapnik, 1995; Haykin, 1998)
Decision trees (C4.5) (Quinlan, 1993)
K-nearest neighbors (*k*-NN) (Aha, Kibler, & Albert, 1991)
Boosted C4.5 (B.C4.5) (Freund & Schapire, 1996)
Logistic regression (Logistic) (Haykin, 1998)

better performance than individual classifiers and text processing approaches.

5. Methodology

To give credibility to the found results and in order to make the experiments reproducible, we detail the experimental methodology as follows.

5.1. Datasets and data representation

We have used nine real English, public and non-encoded datasets. Table 3 summarizes the class distribution and the main theme related with the object of interest. Moreover, each sample is labeled as positive or negative, according to opinion expressed by comments in respect to the object.

We have preprocessed the first four listed datasets in the same way as described in Saif, Fernández, He, and Alani (2013). The last three ones were collected from Twitter and labeled by the authors in the second half of 2014. They are publicly available at http://dcomp.sor.ufscar.br/talmeida/sentcollection/.

Samples were encoded using unigram and numbers, hash-tags, replies and symbols often used in Twitter were kept. In addition, we have created two new attributes to count the amount of positive and negative terms in each sample. Basically, each token in a message is searched in an opinion English lexicon dictionary[6], and if it is associated with a positive value, then the positive attribute is incremented by one, otherwise, if it is associated with a negative value, the negative attribute is incremented by one. According to Mostafa (2013) and Nastase and Strube (2013), similar approaches have presented good results in opinion mining.

5.2. Classification methods

The proposed ensemble system is composed by the well-known and established classification methods available in literature (Table 4). Such approaches are listed by Wu et al. (2008) as the top-performance classification and data mining techniques currently available. We have selected approaches with different hypothesis representation and selection techniques such as probability, optimization, distance and tree-based ones. With such feature, the ensemble besides being flexible can also lead to produce stronger and more generic hypotheses, since it naturally privileges the more adequate strategy for each dataset. Each individual classification technique was also further used to assess the ensemble performance.

[6] The opinion English lexicon dictionary is available at http://goo.gl/czIfkd.

Table 5
Average F-Measure achieved by each evaluated classifier over each dataset.

Methods	Archeage	HCR	Hobbit	IPhone6	OMD	Sanders	SS-Tweet	STS-Test	UMICH
B. C4.5	0.785 ± 0.04	0.685 ± 0.04	0.881 ± 0.04	0.677 ± 0.06	0.743 ± 0.03	0.703 ± 0.03	0.562 ± 0.02	0.807 ± 0.06	0.934 ± 0.02
C4.5	0.753 ± 0.02	0.643 ± 0.03	0.864 ± 0.02	0.681 ± 0.05	0.682 ± 0.04	0.640 ± 0.04	0.543 ± 0.03	0.752 ± 0.06	0.920 ± 0.02
Ensemble	**0.869 ± 0.02**	**0.733 ± 0.03**	**0.921 ± 0.02**	**0.738 ± 0.04**	**0.811 ± 0.02**	**0.756 ± 0.03**	**0.612 ± 0.02**	**0.863 ± 0.03**	**0.969 ± 0.01**
KNN	0.723 ± 0.03	0.633 ± 0.03	0.844 ± 0.04	0.651 ± 0.07	0.728 ± 0.05	0.675 ± 0.04	0.525 ± 0.06	0.776 ± 0.06	0.953 ± 0.02
Logistic	0.836 ± 0.03	0.679 ± 0.03	0.908 ± 0.03	0.698 ± 0.05	0.775 ± 0.02	0.711 ± 0.03	0.585 ± 0.02	0.804 ± 0.03	0.962 ± 0.02
NB B.	0.856 ± 0.02	0.695 ± 0.03	0.869 ± 0.05	0.735 ± 0.03	0.781 ± 0.02	0.723 ± 0.05	0.598 ± 0.02	0.817 ± 0.05	0.948 ± 0.01
NB G.	0.793 ± 0.02	0.590 ± 0.02	0.772 ± 0.08	0.689 ± 0.04	0.628 ± 0.04	0.651 ± 0.02	0.530 ± 0.03	0.751 ± 0.04	0.760 ± 0.06
NB M.	0.843 ± 0.02	0.683 ± 0.03	0.882 ± 0.03	0.699 ± 0.05	0.781 ± 0.04	0.727 ± 0.03	0.593 ± 0.04	0.821 ± 0.05	0.956 ± 0.01
SVM-L	0.842 ± 0.03	0.686 ± 0.03	0.913 ± 0.03	0.708 ± 0.04	0.771 ± 0.01	0.692 ± 0.04	0.583 ± 0.03	0.813 ± 0.03	0.965 ± 0.01
SVM-R	0.820 ± 0.02	0.695 ± 0.02	0.856 ± 0.07	0.651 ± 0.07	0.736 ± 0.05	0.673 ± 0.09	0.539 ± 0.09	0.756 ± 0.07	0.953 ± 0.02

5.3. Evaluation and ensemble system pipelines

First, each dataset was randomly split in 20% of samples for model selection and 80% for training and testing. In model selection stage, the input dataset is expanded by all possible combination of text processing and semantic indexing techniques, resulting in a set of expanded databases (one for each possible distinct merging rule). Then, for each resulting database, a grid search is performed to select good values for the main parameters of each classification method. Such grid is evaluated with F-measure using 5-fold cross-validation. Once good parameters are found, the system determines which expanded dataset performs better for each classification method. In other words, this stage determines which text processing steps are the most suitable for each classification technique. Finally, the constant T used to compute the voting weight of each classifier (Eq. 1) was empirically set to be equal to 0.05.

In classification stage, the biggest part of original input dataset (80% of samples) is used to train the selected model and test it. For this, we randomly split the remaining samples in two parts: training set (75%) and test set (25%). The whole process is repeated ten times with random stratified selection of samples for training and test.

To assess the performance achieved by the ensemble, we have used the same steps for each individual classification method and collected the results for further comparison. For this, to provide a fair comparison, we highlight that the most adequate text processing steps (merging rules) were selected for each individual classification approach and grid search was also performed to set the main parameters, using exactly the same procedure and amount of samples described above.

6. Results

Table 5 presents the average F-measure and standard deviation achieved by each evaluated classifier over each dataset. Bold values indicate the best scores.

The results indicate that, under the same condition and methodology, the proposed ensemble system clearly presented an overall superior performance to any of the other evaluated individual classifier. However, to ensure that results were not obtained by chance, we have performed a statistical analysis using the non-parametric Friedman test (Friedman, 1940) by carefully following the methodology described in Japkowicz and Shah (2011).

The Friedman test checks if the null hypothesis, which states there is no difference between the results, can be rejected based on ranking position of each classifier over each dataset.

The null hypothesis can be rejected if χ^2_F follows a χ^2 distribution with $k-1$ degrees of freedom for number of datasets $(n) > 15$ or number of methods $(k) > 5$. For smaller n and k, the χ^2 approximation is imprecise and a table lookup is advised from tables of χ^2_F values specifically approximated for the Friedman test (Japkowicz & Shah, 2011). For a confidence interval $\alpha = 0.001$, $n = 9$ and $k = 9$, the critical value is 27.877. Given that $\chi^2_F = 63.3515$, we can safe conclude that there is a significant difference between the performance achieved by classification methods and, therefore, the null hypothesis might be rejected.

Next, we have performed the Nemenyi post-hoc test (Nemenyi, 1963) to compare the results pairwise. Such test indicates the performance achieved by the proposed ensemble system differs with high significance ($p < 0.001$) of any other evaluated technique. Therefore, we can safely conclude that its performance is statistically superior than any other individual evaluated approach with a 99.9% confidence level.

As expected, although the ensemble system presents the best overall prediction power, the downside is that it is more time-consuming. Is is mainly because the ensemble combines different text processing techniques and classification methods. However, the bulk of computation time refers to model selection and training, and differences in time for classifying are practically insignificant. Therefore, in scenarios where the most expensive processes can be performed offline, the ensemble system is highly recommended.

7. Conclusion and future work

The task of automatically detecting opinion in short messages posted on social networks is still a real challenge nowadays. Two main issues make difficult the application of established classification algorithms for this specific field of research: the low number of features that can be extracted per message and the fact that messages are filled with idioms, abbreviations, and symbols.

In order to fill these gaps, we proposed an ensemble system that automatically combines the most recommended text processing techniques with established classification methods. For this, we first presented a text expansion method based on lexicographical and semantic dictionaries along with state-of-the-art techniques for semantic analysis and context detection. They were employed to normalize terms and create new attributes in order to change and expand the original text samples aiming to alleviate factors that can degrade performance, such as redundancies and inconsistencies.

We evaluated the proposed ensemble of classifiers with nine public, real and non-encoded datasets. We also performed a statistical analysis on our results, that clearly indicated that the ensemble is statistically superior to any other individual evaluated classification method with a 99.9% confidence level. However, as the proposed approach presented the highest cost in terms of computing power, it is recommended for use in applications where the most expensive processes can be performed offline.

Currently, we are planning to evaluate the presented ensemble in applications with similar characteristics to those presented in this paper, such as content-based short comment filtering. We also intend to apply the proposed system to detect opinions found in messages posted on other Web applications, such as Youtube, forums and blogs. Furthermore, for future work, we aim to (1) parallelize the most expensive computing process to speed up the model selection and training stages, and (2) merge our proposed method with the one suggested in Jiang, Zeng, and Zhang (2013), which combines co-training applied to a multi-classifier system. Both techniques offered improvements on results over individual established classifiers.

Acknowledgment

The authors would like to thank the financial support of Brazilian agencies FAPESP (grant 2014/01237-7) and Capes.

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of 2011 LSM*. In *LSM '11* (pp. 30–38). Association for Computational Linguistics.

Agirre, E., & Edmonds, P. (2006). *Word sense disambiguation*. Springer.

Aha, D., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning, 6*(1), 37–66.

Almeida, T., Almeida, J., & Yamakami, A. (2011). Spam filtering: how the dimensionality reduction affects the accuracy of naive bayes classifiers. *JISA, 1*(3), 183–200.

Analytics, S. (2011). Dataset - Twitter sentiment. http://www.sananalytics.com/lab/twitter-sentiment/. [Online; accessed 07-July-2015].

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., & Aswani, N. (2013). Twitie: an open-source information extraction pipeline for microblog text. In *Proceedings of 2013 RANLP*. In *RANLP'13* (pp. 83–90). Hissar, Bulgaria.

Cook, P., & Stevenson, S. (2009). An unsupervised model for text message normalization. In *Proceedings of the 2009 CALC* (pp. 71–78). Association for Computational Linguistics.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Denecke, K. (2008). Using SentiWordNet for multilingual sentiment analysis. In *Proceedings of 2008 ICDEW* (pp. 507–512). Cancun, Mexico.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of 2000 MCS*. In *MCS '00* (pp. 1–15). Cagliari, Italy.

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. In *Proceedings of the 13rd ICML* (pp. 148–156). Bari, Italy.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of *m* rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92.

Gabrilovich, E., & Markovitch, S. (2005). Feature generation for text categorization using world knowledge. In *Proceedings of the 19th IJCAI* (pp. 1048–1053). Edinburgh, Scotland.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *Technical Report*. Stanfod University.

Gómez Hidalgo, J. M., Buenaga Rodríguez, M., & Cortizo Pérez, J. C. (2005). The role of word sense disambiguation in automated text categorization. In *Proceedings of the 10th NLDB* (pp. 298–309). Alicante, Spain.

Haykin, S. (1998). *Neural networks: A comprehensive foundation* (2nd). New York, NY, USA: Prentice Hall.

Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms - A classification perspective*. Cambridge University Press.

Jiang, Z., Zeng, J., & Zhang, S. (2013). Inter-training: Exploiting unlabeled data in multi-classifier systems. *Knowledge-Based Systems, 45*, 8–19.

Kim, H., Kim, H., Moon, H., & Ahn, H. (2011). A weight-adjusted voting algorithm for ensembles of classifiers. *Journal of the Korean Statistical Society, 40*(4), 437–449.

Kontopoulos, E., Berberidis, C., Dergiades, T., & Bassiliades, N. (2013). Ontology-based sentiment analysis of Twitter posts. *Expert Systems with Applications, 40*(10), 4065–4074.

Luo, B., Zeng, J., & Duan, J. (2016). Emotion space model for classifying opinions in stock message board. *Expert Systems with Applications, 44*, 138–146.

Maynard, D., Bontcheva, K., & Rout, D. (2012). Challenges in developing opinion mining tools for social media. In *Proceedings of 2012 LREC*. In *LREC'12*. Istanbul, Turkey.

Mostafa, M. M. (2013). More than words: Social networks' text mining for consumer brand sentiments. *Expert Systems with Applications, 40*(10), 4241–4251.

Nastase, V., & Strube, M. (2013). Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence, 194*(1), 62–85.

Navigli, R., & Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(4), 678–692.

Navigli, R., & Ponzetto, S. P. (2012). Multilingual WSD with just a few lines of code: the BabelNet API. In *Proc. of 2012 ACL*. In *ACL '12* (pp. 67–72). Jeju Island, South Korea.

Nemenyi, P. F. (1963). *Distribution-free multiple comparisons*. Princeton University Ph.D. thesis.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Fundations and Trends in Information Retrieval, 2*(1–2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of 2002 ACL EMNLP*. In *ACL EMNLP '02* (pp. 79–86). Philadelphia, USA.

Quinlan, J. (1993). *C4.5: Programs for machine learning* (1st). San Mateo, CA, USA: Morgan Kaufmann.

Saif, H., Fernández, M., He, Y., & Alani, H. (2013). Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. *1st interantional workshop on emotion and sentiment in social and expressive media: Approaches and perspectives from AI (ESSEM 2013)*.

Shamma, D. A., Kennedy, L., & Churchill, E. F. (2009). Tweet the debates: understanding community annotation of uncollected sources. In *Proceedings of 2009 WSM*. In *WSM '09* (pp. 3–10). ACM.

Speriosu, M., Sudan, N., Upadhyay, S., & Baldridge, J. (2011). Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of 2011 EMNLP*. In *EMNLP '11* (pp. 53–63). Association for Computational Linguistics.

Taieb, M. A. H., Aouicha, M. B., & Hamadou, A. B. (2013). Computing semantic relatedness using wikipedia features. *Knowledge-Based Systems, 50*(9), 260–278.

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology, 63*(1), 163–173.

UMICH (2011). *Dataset SI650 - Sentiment Classification*. https://goo.gl/Xfr8lI. [Online; accessed 07-July-2015].

Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. (2014). Sentiment classification: the contribution of ensemble learning. *Decision Support Systems, 57*, 77–93.

Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., et al. (2008). Top 10 algorithms in data mining. *KAIS, 14*(1), 1–37.

Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences, 181*(6), 1138–1152.

Xue, Z., Yin, D., Davison, B. D., & Davison, B. (2011). Normalizing Microtext. In *Proceedings of the 2011 AAAI* (pp. 74–79). Association for the Advancement of Artificial Intelligence.