Applied Mathematics and Computation 270 (2015) 756–767

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A polarity analysis framework for Twitter messages

Ana Carolina E.S. Lima [a,b,*], Leandro Nunes de Castro [a], Juan M. Corchado [b]

[a] Natural Computing Laboratory, Mackenzie Presbyterian University, São Paulo, Brazil
[b] Department of Computer Science, University of Salamanca, Salamanca, Spain

ARTICLE INFO

Keywords:
Social media
Twitter
Sentiment analysis
Text mining
Machine learning
Classification task

ABSTRACT

Social media, such as Twitter and Facebook, allow the creation, sharing and exchange of information among people, companies and brands. This information can be used for several purposes, such as to understand consumers and their preferences. In this direction, the sentiment analysis can be used as a feedback mechanism. This analysis corresponds to classifying a text according to the sentiment that the writer intended to transmit. A basic sentiment classifier determines the sentiment polarity (negative, neutral or positive) of a given text at the document, sentence, or feature/aspect level. Advanced types may consider other elements like the emotional state (e.g. angry, sad, happy), affective states (e.g. pleasure and pain), motivational states (e.g. hunger and curiosity), temperaments, among others. In general, there are two main approaches to attribute sentiment to tweets: based on knowledge; or based on machine learning algorithms. In the latter case, the learning algorithm requires a pre-classified data sample to determine the class of new data. Typically, the sample is pre-classified manually, making the process time consuming and reducing its real time applicability for big data. This paper proposes a polarity analysis framework for Twitter messages, which combines both approaches and an automatic contextual module. To assess the performance of the proposed framework, four text datasets from the literature are used. Five different types of classifiers were considered: Naïve Bayes (NB); Support Vector Machines (SVM); Decision Trees (J48); and Nearest Neighbors (KNN). The results show that the proposal is a suitable framework to automate the whole polarity analysis process, providing high accuracy levels and low false positive rates.

1. Introduction

Social media, such as online social networks, blogs, microblogs and collaborative projects, have become a way of expressing collective interests. People are motivated by the sharing of information and the receipt of feedback from friends and colleagues. The volume of data generated by these sites allows unprecedented investigations about how societies are organized [1–5].

Among the many social media sites, Twitter is a popular microblog service, founded in 2006, designed for simplified (short messages) communication, thus accelerating the process of message update. There are about 250 billion messages posted daily in Twitter, approximately 100,000 messages per minute [6], characterizing this service as an important repository for data analysis, and useful source of content for marketers, psychologists and others interested in the extraction and mining of opinions, views, moods and attitudes [7].

* Corresponding author. Tel.: +55 11 2114 8503.
E-mail addresses: aceslima@gmail.com (A.C.E.S. Lima), lnunes@mackenzie.br (L.N. de Castro), corchado@usal.es (J.M. Corchado).

http://dx.doi.org/10.1016/j.amc.2015.08.059
0096-3003/

Many investigations are being performed around Twitter, such as sentiment analysis [8,9], event detection [10,11], online reputation analysis [12], trends identification [13,14], stock trends [15], crowd size [16], regional differences [17] and others. Companies see social media services as an important place to monitor and promote their brands, with one basic principle: *if something is said in the social media, then it can be qualified and quantified* [18]. Quantitative measures can, for instance, provide information about the impact and dissemination of a message and the frequency with which a subject is mentioned, besides the sentiment associated with the text. On the other hand, qualitative measures are made by teams of analysts according to the goals, needs and characteristics of the project [19].

One of the feedback mechanisms most used in Twitter data analysis is the *sentiment analysis*, which provides a view about the sentiment expressed in the messages [20]. This sentiment is basically labeled according to the text polarity, that is, whether the message has a positive, negative or neutral connotation. For companies, this measure helps to observe the public and market opinion about itself [21]. This task is named *polarity analysis* or *sentiment polarity* [22].

Polarity determination can be made at different levels: document [23], sentence [24], word [25], or attribute [26]. The document level considers the whole document as the basic unit. The sentence level extracts and determines the sentiment in each subjective sentence of the text. In the word level, each word in the text is analyzed and classified. The attributes' level identifies and extracts attributes of an entity (e.g., product, person, company, etc.) on text and determines an opinion for each attribute [27].

There are two main approaches to building a framework for polarity analysis: based on lexical dictionaries (knowledge-based); or based on machine learning algorithms. The first approach when designed for Twitter text makes the system vulnerable to the short messages, informal language with slangs and swear words, absence of explicit sentiments, presence of special characters, mixed language, among other features peculiar to tweets. On the other hand, to build a system based only on machine learning algorithms make it dependent upon the rapid creation of a training set.

The purpose of this paper is to introduce a polarity analysis framework focused on Twitter messages, which combines both approaches. This framework uses techniques specifically designed to deal with short messages, such as tweets. The classification is performed by a two-stage machine learning approach, which includes an automatic knowledge-based classifier and the machine learning algorithms. This framework provides a modular framework in which each module has different approaches that can be configured according to the application domain. To evaluate our proposal four Twitter datasets from the literature were used: Debate 2008 [28], SentiStrenght [29], Sanders [1] and SemEval 2013.[2]

The paper is organized as follows. Section 2 provides an overview of the polarity analysis problem, and Section 3 reviews related works. Section 4 describes the proposed framework, and Section 5 illustrates how the proposed framework is employed. Section 6 presents and discusses the obtained results. The paper is concluded in Section 7 with a general discussion about the proposal and avenues for future investigation.

2. Polarity analysis: an overview

It has been long since people seek and observe the opinions of others to direct their behaviors, such as purchases or considerations about a determined subject [30]. The habit of questioning friends and family members before purchasing a product or service precedes the web, but with the proliferation of social media sites, such as microblogs and social networks, people started sharing their experiences and opinions over the Internet. However, finding a source of opinion, monitoring and analyzing it, requires a lot of work, often unfeasible manually. Thus, the need to automate this process arises and *sentiment analysis* emerges as an area of active research. Sentiment analysis, also called *opinion mining*, aims at understanding how a reader can interpret the subjectivity within a text and then transpose it to an algorithm that can perform this task automatically. In natural language the subjectivity concepts involve aspects of language used to express opinions, evaluations, sentiment, speculations and emotion [31].

Most authors accept a simplified representation of sentiment according to its *polarity*. The *sentiment polarity* can be understood as a point on a rating scale that corresponds to the positive or negative evaluation of the significance of this sentiment. The sentiment classification, in turn, examines the *polarity* of a subjective text [27,5]. For example, given a review about a product, the sentiment analysis system determines whether the sentiment expressed in the review has a positive or negative connotation.

Typically, text polarity can be *positive*, *negative* and, eventually, *neutral* for determining the sentiment absence in the text. This classification problem is defined as follows: given a document $\mathbf{d}_i \in \mathbf{D}$, $\forall i$, it is associated with a class belonging to the set $C = \{c_1, c_2, \ldots, c_k\}$, also called labels or categories. Through a learning method, or learning algorithm, the classifier learns a function γ that maps each document into one of the classes: $\gamma: \mathbf{D} \rightarrow C$ [32]. This classification can be divided in two major cores: *subjectivity detection*; and *sentiment classification*. Subjectivity is focused on determining whether a text or sentence in this text has subjectivity (opinion, emotions, evaluations, beliefs or speculations), or whether it is merely a fact, while sentiment classification aims to determine a label for this subjectivity [31]. This subjectivity detection prevents the sentiment classifier to consider some texts that are irrelevant or potentially deceptive. Moreover, it reduces the dimensionality of the set of labels that can be assigned to a text, that is, at first it is checked whether a text is objective or subjective, and those considered subjective go through a new classification process that will determine the polarity. The objective ones are labeled as neutral.

[1] http://www.sananalytics.com/lab/twitter-sentiment/.
[2] http://www.cs.york.ac.uk/semeval-2013/.

Besides these two subtasks the sentiment analysis could also involve the identification of the target entity, that is, which object in the text the sentiment is directed to and the identification of the person or organization that expresses the opinion [33].

There are texts in which a polarity is not so remarkable, occurring mixed sentiments, that is, when both positive and negative remarks are made, which is a different case than a neutral text which is only factual and has already been addressed in the literature. In this case, the polarity analysis has four labels: positive, negative, neutral and mixed. In the experiments to be performed here the "mixed" label is discarded.

2.1. Polarity analysis main approaches

Works on polarity analysis can be divided into two approaches: *lexicon-based* (also called knowledge-based), and *machine learning-based*. The lexicon-based approaches make use of dictionaries, like Linguistic Inquiry and Word Count (LIWC) [34] and SentiWordNet [35], to determine subjectivity. Usually, a dictionary is formed by words and a corresponding classification value, for example, a dictionary may contain the word *enjoy* and the value $+1$ indicating that this word has a positive polarity, or, by contrast, it may contain the word *hate* with value -1 indicating a negative polarity. The simplest form of obtaining the text polarity is to sum the values of all words present in the text and determine the resulting polarity: if the sum is positive, then the text has a positive polarity; otherwise, the text receives a negative polarity. The main problems with the lexicon approach are the low coverage of the words of the dictionary in relation to the whole database and the specificity of the dictionary. There are two alternatives in such situations: to discard the messages not covered by the dictionary; or to use a different classifier for such messages.

The machine learning approaches use learning algorithms, such as Naïve Bayes (NB), maximum entropy (ME), and support vector machines (SVM), to assign polarity. These methods are normally used to perform text classification and have shown effectiveness when applied to the polarity and sentiment analysis problems [36–38,14,39]. In [40] the authors showed that machine learning techniques applied to sentiment analysis produce better results when compared with those obtained by random choice (50%), or those made by human classification (between 58% and 64%). This approach requires a pre-classified database sample, called *training set*, which is either used to generate a classifier (classification model) or to compare with new unlabeled data to be classified. This is important because the classifier accuracy is highly dependent upon such training data. Besides, it is necessary a database sample with unknown labels, called test set, to evaluate the generalization capability of the algorithm to new data [41]. The best results presented by these techniques are directly related to the quality of the training set. Normally, the pre-classification of the training set is made manually, which makes the work hard and subject to different human perceptions. Also, when the application involves social media data, manual classification becomes unfeasible due to the volume and velocity of the data generated continuously.

2.2. Hybrid approaches

In many cases, lexicon-based and machine learning approaches are combined in order to solve the lexicon-based approach coverage problem and to reduce the training set formation dependency of the machine-learning approaches. In [42] it was proposed the automatic generation of training data by taking into account the sentiment available in texts containing *emoticons*. Emoticons are graphic representations formed by punctuation marks and letters representing facial expressions, generally included as a means to express the mood of a person. The author collected a corpus of texts, from the Usenet newsgroup, marked with emoticons and extracted the paragraphs containing the emotion of interest from each message, removing any superfluous formatting characters. For the test set it were used 2000 articles containing smiles and 2000 containing frowns; and for the training set 20,000 articles were considered. The classifiers used were the Naïve Bayes, which achieved a mean accuracy of 61.5%, and the SVM, which achieved a mean accuracy of 70.1%.

In [37] the authors created a training set from tweets with emoticons and applied Naïve Bayes, maximum entropy and SVM as sentiment classification algorithms. The training set was composed of 800,000 tweets with positive emoticons and 800,000 tweets with negative emoticons, while the test set was composed of 117 negative tweets and 182 positive tweets, both manually marked. In the classification phase all emoticons were removed from the messages, because the SVM and Maximum Entropy classifiers would give a high weight to these characters. The three classifiers achieved a mean accuracy of around 80%.

The study of [4] showed that the fraction of tweets with emoticons may not exceed 10% of the whole database. For this study the authors analyzed all messages produced between 2006 and 2009 on Twitter, the microblog start date, a total of 1.1 billion tweets. In [29] the authors used machine learning with LIWC to assign the sentiment strength of a text. LIWC is a commercial tool for text analysis that estimates emotional, cognitive and structural components of a text. Each word has a category and weight. In addition, the authors have added an extra set of words, a set of emoticons and repeated punctuation according to the context of social media. The evaluation of the technique was performed in six different bases: MySpace, Twitter, Digg, BBC forum, Runners World and YouTube [43]. The sentiment polarity could be binary (positive/negative) or ternary (negative/neutral/positive) and associated with strength in the range $[-4, 4]$ [29].

In [44] the authors used emoticons and lists of the most commonly used positive and negative words provided by Twitrratr [3] with the Naïve Bayes algorithm. Moreover, the paper introduced the *Polarity Score Technique*, which allows the determination of

[3] http://twitrratr.com/.

the polarity score at the level of individual words of a tweet. For example, the word 'happy' is used predominantly for expressing the positive sentiment, with this method a popularity factor (pF) is multiplied by the score of each unigram token which has been scored in the previous steps. The performance was evaluated with the Stanford [37] and Mejaj datasets. The Stanford data contains 1.6 million tweets with emoticons in the training set with an equal number of positive and negative messages; and 438 tweets (180 positive, 180 negative and 130 neutral) in the test set. The Mejaj dataset contains 1,464,638 tweets in the training set (668,975 positive and 795,661 negative) and 402 tweets in the test set (198 positive and 204 negative).

Silva et al. [45] proposed a classification approach which combines classifier ensembles and lexicons (Hu and Liu word list and emoticons). The Hu and Liu list contains 4783 negative words and 2006 positive words. Ensemble methods combine multiple classifiers to generate a single classifier. They used Multinomial Naive Bayes, SVM, Random Forests, and Logistic Regression. To represent Twitter messages in vector space the authors compared bag-of-words and feature hashing. The bag-of-words was constructed with binary frequency. Retweets, stop words, links, URLs, mentions, punctuation, and accentuation were removed so that data set could be standardized. Stemming was performed so as to minimize sparsity. The classifier ensembles were obtained from the combination of lexicons, bag-of-words, emoticons, and feature hashing. They used the follow datasets: Sanders, Stanford, Obama-McCain Debate and Health Care Reform (HCR).

This paper proposes a framework to classify the sentiment of Twitter messages according to their polarity. We combine emoticons and explicit declarations of mood/emotions to generate the training set for a sentiment analyzer and then develop a framework to automatically perform polarity classification of Twitter messages. In a similar work, entitled TOM [46], the authors proposed a framework infrastructure to polarity classification of tweets (positive, negative or neutral) based on the core idea of preprocessing the messages. They proposed removing the slangs, grammatical mistakes, abbreviations and other noise and then feed it to the classifier. The proposed system is composed of three main modules: data acquisition; pre-processing and transformation of tweets containing real-valued features; and the application of different classification techniques, in a hierarchical way. The classifier hierarchy is as follows: emoticon classification; polarity classification by list of positive and negative words; and SentiWordNet classification, in which, the classification is performed by SentiWordNet dictionary. To evaluate the system they collected 2.116 tweets with Twitter4J and classified them manually. Finally, the authors compared the results of TOM with emoticon classification, word list classification and SentiWordNet classification separately. When compared with TOM, our proposal is broader, because besides preprocessing text with the removal of URLs, mentions, and hashtags, it uses knowledge-based dictionaries to perform the automatic classification based on context and machine learning algorithms. The next section details our proposed technique.

3. A polarity analysis framework for Twitter messages

This paper proposes a *Polarity Analysis Framework*, PAFRA for short, which combines machine learning and lexicon-based approaches so as to provide an automatic and accurate polarity classification of Twitter messages. The main features of the proposed framework are:

- The combination of lexicon-based and machine-learning based approaches in a single scheme to perform polarity analysis;
- The automatic generation of the training set for the machine-learning approaches;
- The focus on short messages with automatic classification by contextual verification;
- The incorporation of multiple techniques, such as entity detection, to improve the accuracy and reduce the false negative rate, a recurring problem in polarity and sentiment analysis.

The proposed framework takes into account the following definitions:

Definition 1. Let ω be the set of *words* from the knowledge base associated with a given polarity, named *polarized word*. Now, define s_ω as the *support* of set ω; that is, s_ω is the percentage of documents that contain at least one term from the set ω.

Definition 2. Let ε be the set of *emoticons* from the knowledge base associated with a given polarity. Now, define s_ε as the *support* of set ε; that is, s_ε is the percentage of documents that contain at least one term from the set ε.

Definition 3. Let a *classifying element* (CE) be any emoticon or word that provides an explicit polarity to a tweet. The CE may be positive (CE_+), negative (CE_-) or neutral ($CE_\#$) in accordance with the polarity attributed. If the tweet contains a CE, then it is processed by an automatic classifier. Otherwise, it is processed by a machine-learning algorithm. The tweets processed by the automatic classifier are used to form the training set for the machine-learning approach.

Definition 4. Let a *classification target* (CT) be any object or subject for which the polarity has to be assigned. In this version the CT is provided by the user before running the classifier. This definition is important because one challenging aspect in polarity analysis is that a sentiment can be expressed over anything. Therefore, it is important to identify the object (or subject) that one desires to know the opinion about. To illustrate, consider the following message: "*although the screen is small, the television is amazing*". If the goal is to understand what was the sentiment of the writer about the screen, the object to be analyzed is the screen and the polarity is negative. By contrast, if the goal is to understand the sentiment about the television as a whole, the object to be analyzed is the television and the polarity is positive. Usually, when one desires to perform polarity analysis the target item (e.g. object, brand, person, etc.) is known in advance and, thus, the CT can be user-defined based on context.

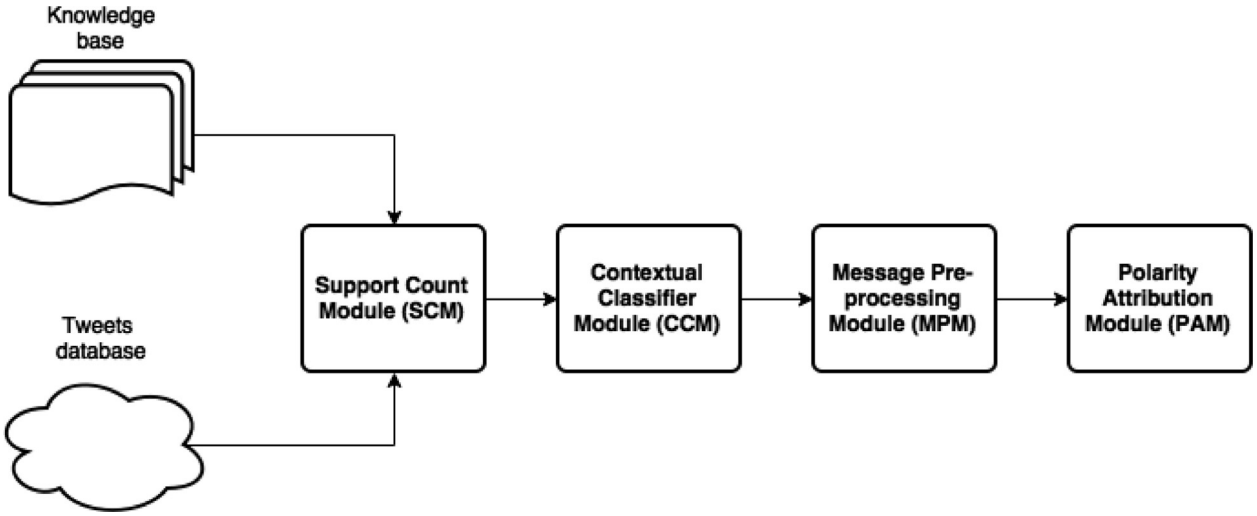

Fig. 1. Framework modules.

Taking into account the above definitions, the framework is composed of two different databases (tweets database and knowledge base) and four modules, as illustrated in Fig. 1: *support counting module* (SCM); *contextual classification module* (CCM); *message pre-processing module* (MPM); and *polarity attribution module* (PAM). Each of these modules will be detailed in the following sections.

3.1. Knowledge and tweet bases

The knowledge base used is the union of seven lexical databases:

1. Emoticons: constructed from the Yahoo Message,[4] MSN Message,[5] Gtalk Message,[6] and SentiStrenght [7] emoticons lists;
2. Affective Norms for English Words (ANEW) [47];
3. Positive and Negative Affect Scale-Extended (PANAS-X) [48];
4. SenticNet [49];
5. LIWC;
6. Sentiment Strength; and
7. SentiWordNet.

This combination creates a knowledge base from the list of words that appear in at least two of the seven lexicons. Potential conflicting polarity of repeated words is resolved by a majority voting scheme. The tweets base corresponds to the messages that will have their polarity inferred.

3.2. Support counting module (SCM)

This module is responsible for checking the percentage of tweets that contain at least one polarized word from the set ω or one emoticon from the set ε. The premise here is that the sentiment-based words (emoticons) must have a minimal support, $\min\_s_\omega$ ($\min\_s_\varepsilon$); that is, a minimal coverage of the set of documents in order to serve as labels to train the machine-learning algorithms. In the experiments to be performed here to illustrate and validate the framework it will be assumed $\min\_s_\omega = 5\%$ or $\min\_s_\varepsilon = 5\%$.

3.3. Contextual classification module (CCM)

In principle, the proposed framework defines the class of the messages by means of the first occurrence of the classifier element (CE). Then, it performs a check that verifies the proximity between the CE and the classification target (CT). The framework then deals with the occurrence of multiple CEs so as to identify which one is directly related to the CT. This checking system analyses the proximity between the CE and the CT, and verifies if there is more than one classifying element in the text. It works based on the following rules:

[4] http://messenger.yahoo.com/features/emoticons.
[5] http://messenger.msn.com/Resource/Emoticons.aspx.
[6] https://code.google.com/p/emoticony/wiki/Emoticons.
[7] http://sentistrength.wlv.ac.uk/.

- Tweet with CE_-: it is verified if the CE_- is next to the CT, if yes, then the message is classified as negative. For example, "I don't like football, but this game is ok". Assume that the goal is to assess the sentiment of the message in relation to the game. In this case, the message contains the CE, but it is far from the CT ('this game'). Therefore, without the verification the message would be classified as negative; however, the CE does not make reference to the CT but instead to football, and thus the message will not be classified as negative.
- Tweet with CE_+: it is verified if the CE_+ is next to the CT and if there is no polarity reversing word before the element, such as 'no', 'never', etc. If it is next to and there is no reversal word, the message is classified as positive. For example: "I didn't like this program". In this case, the CE_+ ('like') is next to the CT, but there is a reversal word before the classifier element and the message is not classified as positive.
- Tweet with CE_+ and CE_-: it is verified which element is closest to the CT. The sentiment is defined by the closest one. For example: "This soccer is awful, I want the great 'the walking dead'". In this case there is the CE_- and CE_+. The system verifies which one is the closest to the CT to assign the correct class. In this case, the tweet is classified as positive if the classification target is the television program 'the walking dead', and negative if CT is the soccer game.
- Tweet with $CE_\#$: if the tweet contains only the neutral classifier element, then this is the class assigned to the message.
- Tweet without a CE: if a tweet contains no classifier element, it will be classified by the sentiment attribution module.

The automatic classification of tweets includes feature selection, dataset stratification, and change of emoticons by keywords. The feature selection aims to reduce the dimension of the data matrix by the elimination of attributes with little relevance, improving the classifier performance and reducing preprocessing time. The objective of stratification in an automatically classified database is to ensure a number of objects of each training class proportional to the number of objects in each class of the original dataset. Finally, text preprocessing steps ignore repeated words or special characters, thus, the emoticons are eliminated after the conversion of the tweets into a data matrix. To avoid this, a change of emoticons to pre-defined sentiment words was implemented, as follows: positive emoticons were changed by the word "happy"; while negative ones were replaced by the word "sad". After the contextual classification, tweets are divided into training and testing. The training set contains the tweets automatically classified by the knowledge-base, and the test set contains the unlabeled tweets that have to be classified by a machine-learning approach.

3.4. Message pre-processing module (MPM)

Generally, natural language texts cannot be directly processed by classifiers and learning algorithms. Thus, the pre-processing module converts the tweets into a representation that is manageable by the classifiers. Typically, texts are represented by feature vectors, and the most common approach is to transform a text into a 'bag-of-words' model. In the standard representation, each word in a message becomes a feature of a vector, and the dimension of the feature space is equal to the number of different words in the collection. We explore the following representations:

- **N-gram:** contiguous sequence of n words, forming n-grams. In this case, each n-gram is one feature of the feature space whose dimension is equal to the number of n-grams. When $n = 1$ there is the original case where each word represents a feature. A weight value was associated with each pair $\langle$message, n-gram$\rangle$ using the TF-IDF scheme [50].
- **LIWC:** the LIWC (Linguistic Inquiry and Word Count) is a text analysis program [51] composed of four general descriptor categories (total word count, words per sentence, percentage of words captured by the dictionary, and percentage of words longer than six letters), 22 standard linguistic dimensions (e.g., percentage of words in the text that are pronouns, articles, auxiliary verbs, etc.), 32 word categories tapping psychological constructs (e.g., affect, cognition, biological processes), 7 personal concern categories (e.g., work, home, leisure activities), 3 paralinguistic dimensions (assents, fillers, nonfluencies), and 12 punctuation categories (periods, commas, etc.). [52,34]
- **MRC:** the MRC2 (MRC Psycholinguistic Database) is a machine usable dictionary containing 150,837 words with up to 26 linguistic and psycholinguistic categories.
- **Stanford POS Tagger**: named here sTagger, originally written by Kristina Toutanova [53], reads a text in some language and assigns parts of speech to each word (and other tokens), such as nouns, verbs, adjectives, etc. The English taggers use the Penn Treebank tag set [54].
- **Apache OpenNLP**[8]: named here oNLP, is a machine learning based toolkit for the processing of natural language text; the POS tagger is also based on the Penn Treebank tag set and uses WordNet for lemmatization. WordNet® is a lexical database of English inspired by psycholinguistic theories of human lexical memory, where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept [55].

3.5. Polarity attribution module (PAM)

This module is responsible for attributing polarity to, i.e. classifying, those tweets whose labels are still unknown. It takes the information from the MPM, uses the training data to train a specific machine-learning algorithm, and labels the remaining data based on the classification performed by the algorithm trained with the training data.

[8] https://opennlp.apache.org/.

Table 1
Summary of Twitter datasets used.

Database	Atributes	Polarity labels	Example
OMD	Text, tweet identification, date, user and a sentiment set.	Positive, negative, mixed and other.	Watching by myself #tweetdebate Not drinking :(waiting to start cringing at McCains blunders
SS-Twitter	Text, positive average strength, and negative average strength	Positive and negative strength. Polarity was assigned according to the highest strength	?RT @justinbiebcr: The bigger the better....if you know what I mean;)
Sanders	Tweet identification and polarity label	Positive, negative, neutral or irrelevant	Now all @Apple has to do is get swype on the iphone and it will be crack. Iphone that is
SemEval	Tweet identification and polarity label	Negative, neutral or positive	Gas by my house hit $3.39!!!! Im going to Chapel Hill on Sat. :)

3.6. Illustrating the operation

To illustrate how to use the proposed Polarity Analysis Framework, consider the following example: document set (tweets): D = {"Reading my kindle2... Love it... Lee childs is good read", "@mikefish Fair enough. But I have the Kindle2 and I think its perfect :)", "Kindle2 is launched"}, CT = "Kindle2", ε = {':)', ':(', '=D', '=['} and ω = {'love', 'hate', 'worst', 'excellent'}.

- *SCM*: the first step is to verify if this corpus satisfies the minimum support defined mim_s$_\omega$ = 5%. If the minimum support is greater than 5%, the dataset passes to the SCM; otherwise, it is passed to a manual classification system. Note that the minimal support was chosen empirically, and different values could have been used depending on each application.
- *CCM*: in the contextual classification module each message is analyzed in terms of its context.
  - Document 1 (d1): Message: "Reading my kindle2... Love it... Lee childs is good read" and T = {read, kindle2, love, lee, childs, good, read}. The system identifies the CT 'Kindle2' and classifies this message as positive due to the word 'love'.
  - Document 2 (d2): Message: "@mikefish Fair enough. But I have the Kindle2 and I think its perfect :)" and T = {fair, enough, kindle2, think, perfect, :)}. Again, the system identifies the CT 'Kindle2' and classifies this message as positive due to the emoticon ':)'.
  - Document 3 (d3): Message: "Kindle2 is launched" and T = {kindle2, launched}. Although, the system identifies the CT 'Kindle2' no word or emoticon is found. So, the message is not automatically classified.
- *MPM*: in the third step, the data is pre-processed preparing it to be classified and the database is divided into training and testing. In this case, d_1 and d_2 are used as the training set and d_3 becomes the test set.
- *PAM*: in the final step of the framework, the classified messages are used as training set, while the unlabeled messages are classified by the machine-learning algorithm trained with the classifier set. In the present example, only message d_3 requires the classification based on a learning algorithm.

4. Performance evaluation

This section presents the materials and methods used to assess the performance of the proposed automatic sentiment classifier. Results are then presented and discussed.

Databases – To assess the performance of PAFRA, four Twitter datasets available in the literature were used, as summarized in Table 1:

- **Obama-McCain Debate (OMD)**: consists of tweets about the presidential debate held in the United States in the 2008 campaign. It contains 3238 tweets collected in 27/09/2008 between 01:01 and 03:30. The base is presented in the papers by [28] and by [56]. The dataset attributes are: text, tweet identification, date, user and a sentiment set. The sentiment set is formed by four labels (positive, negative, mixed and other), where each label has the number of votes they received to represent the sentiment of that text. The votes were assigned by Amazon Mechanical Turk (AMT), a crowd-sourcing site where workers complete short tasks for small amounts of money [28]. In the present paper case, polarity is assigned based on a majority vote scheme.
- **SentiStrenght Twitter Dataset (SS-Twitter)**: this is a dataset provided by the developers of this tool. The official website of the tool provides six databases with 4242 tweets. There is no description of the collection period or which subjects were used. The attributes of the dataset are text, positive average strength, and negative average strength. As the dataset comes from the SentiStrenght system, which evaluates the strength of the sentiment, the text polarity is not shown. In the evaluation to be performed here, polarity was assigned according to the highest strength.
- **Sanders**: developed by Nike Sanders, the dataset has 5513 tweets whose sentiments were attributed manually. The dataset contains three items: subject (query), sentiment and tweet id. The dataset does not provide the text, so it is necessary to implement a crawler to access the Twitter API and retrieve this information, or to use the Python code provided by the author. The sentiment can be positive, negative, neutral or irrelevant. The author did not consider the emoticons at the moment of manual assignment of sentiment. Thus, many texts that have positive or negative emoticons were labeled as neutral or irrelevant. For this dataset, we also do not use the emoticons in the experiments to be performed.

Table 2
Proportion of positive and negative tweets after transforming all databases into a binary problem.

Database	Total	Positive	Negative
OMD	2007	743	1264
SS-Twitter	1008	483	525
Sanders	4242	3293	949
SemEval	5083	3573	1510

- **SemEval**: is an event focused on evaluating semantic analysis systems. The 2013 edition presented some competitions, among them the sentiment analysis of Twitter messages. With that it was made available a training dataset with 9684 tweets, and a test dataset with 1654 tweets. The tweets were classified as negative, neutral or positive. As for the Sanders dataset, the message texts were not provided, and a crawler to access the Twitter API and retrieve the text information had to be implemented.

Knowledge and Tweet Bases – The knowledge base used is the union of seven lexical databases:

- Emoticons: constructed from the Yahoo Message,[9] MSN Message,[10] Gtalk Message [11] and SentiStrenght [12] emoticons lists;
- Affective Norms for English Words (ANEW) [47];
- Positive and Negative Affect Scale-Extended (PANAS-X) [48];
- SenticNet [49];
- LIWC;
- Sentiment Strength; and
- SentiWordNet.

This combination creates a knowledge base from the list of words that appear in at least two of the seven lexicons. Potential conflicting polarity of repeated words is resolved by a majority voting scheme. The tweets base corresponds to the messages that will have their polarity inferred.

Preprocessing: We lower-cased all words in the tweets, remove all url, omit the hashtag symbols (e.g., #tweetdebate → tweetdebate) and the user mention symbols (e.g. @justinbiebcr → justinbiebcr).

Machine learning classifiers: The algorithms selected for the classification module were: Naïve Bayes (NB); a support vector machine (SVM); a decision tree (J48); and the K-nearest neighbors (KNN). All algorithms used are machine-learning classification algorithms available in Weka [57] and use the same methodology adopted in [29].

Evaluation metrics: Although the datasets have different labels, all of them contain messages with the positive and negative labels. Thus, only these labelled messages were used for assessment, turning the problem into a binary classification task (Table 2). The performance measures used to evaluate the classifiers were: Precision, Recall and F-measure. These measures are used to evaluate how satisfactory are the answers retrieved by an information retrieval system and, thus, suits the purposes of this research. Precision corresponds to the proportion of the predicted positive cases that were correctly classified, and Recall is the proportion of positive cases that were correctly identified. F-measure is the harmonic mean between Precision and Recall [58]. To measure the overall classifier performance, the Accuracy of the classifier was calculated. It represents the success rate of the classification algorithm and corresponds to the number of correct classifications divided by the number of posts [59]. In addition to the accuracy, the *false positive rate* – FPR (Eq. (5)) corresponds to the rate of incorrect classifications made by the algorithm [59].

4.1. Results and discussion

This section presents the results obtained by applying the proposed framework to the five datasets listed. We evaluate our system in two steps:

- **Contextual classification module:** we observed the quality of the training set produced in this module by the support (coverage) of the knowledge bases for each dataset and the accuracy of the automatic classification algorithm.
- **Polarity attribution module:** this module was tested with different configurations of data attribute representations and several machine-learning classification algorithms available in Weka.

Table 3 shows the support (coverage) of the knowledge bases for each dataset and the accuracy of the automatic classification algorithm. The average knowledge base support (i.e., support of $\{\varepsilon\}\cup\{\omega\}$) for the four datasets was 11.53%, and the average

[9] http://messenger.yahoo.com/features/emoticons.
[10] http://messenger.msn.com/Resource/Emoticons.aspx.
[11] https://code.google.com/p/emoticony/wiki/Emoticons.
[12] http://sentistrength.wlv.ac.uk/.

Table 3
Assessment of the automatic polarity classification algorithm.

Base	OMD (%)	Sanders (%)	SS-Twitter (%)	SemEval (%)
Support	13.20	17.06	7.19	8.66
Accuracy	73.21	82.56	84.59	91.16

Table 4
Distributions of classes in the training and test sets.

	Training		**Test**	
	+	−	+	−
OMD	146	74	1150	637
Sanders	69	55	420	464
SS-Twitter	341	250	2859	792
SemEval	702	328	2790	1263

accuracy of the automatic classifier was 82.88%. For the application, it is assumed that the knowledge base has to have a minimal support of 5%. As all datasets have a support greater than that value, no manual labeling was required.

By investigating the divergences between the automatic classification using only the knowledge base and the polarities originally assigned to the messages, it was possible to observe that, in most cases, users wrote words with a given sentiment but demonstrated another with an emoticon in their messages, what is ambiguous from a sentiment analysis perspective. As our framework gives priority to the emoticons, the divergences in classifications appeared. This problem is more evident in the OMD dataset, and less evident in SemEval.

Table 4 shows the distributions of the training and test sets after the automatic classification and dataset division. OMD and SemEval have more positive tweets, with an average ratio of 65.36% and 68.50% of positive tweets in the training and test set, respectively. The Sanders dataset has the more balanced training and test sets. The SS-Twitter has reasonably balanced classes for training, but many more posts for testing the positive class than the negative one.

The training sets were formed using a stratified approach, balancing the number of positive and negative tweets for all sets of data. The positive class is assumed to be composed of the tweets with positive sentiment. Two sets of experiments will be presented:

- The first one with the data represented using the standard TF-IDF method with n-grams, $n = \{1,\ldots,6\}$; and
- The second one with the attributes based on the dictionary categories.

The polarity attribution module was tested with several standard machine-learning classification algorithms available in Weka [57] and using the same methodology adopted in [29]. The algorithms used for classification were: Naïve Bayes (NB); a support vector machine (SVM); a decision tree (J48); and the K-nearest neighbors (KNN). Besides, in the message pre-processing module, Weka functions were employed to prepare the data. More specifically, the words' frequency was determined using the standard TF-IDF value and the pre-processing was performed as explained in Section III.D.

The results are presented in Table 5 in relation to the accuracy (ACC), false positive rate (FPR), and F-measure. The n-grams used ranged from $n = 1$ to $n = 6$, and the number of k neighbors chosen for k-NN was 3. For the OMD dataset, the Naïve Bayes algorithm achieved the best performance with a 6% FPR and 63.01% accuracy. Although J48 presented a slightly superior performance in terms of accuracy, its false positive rate was very high, indicating that the negative class was almost entirely misclassified. The usually low accuracies are a consequence of the difficulty in identifying the positive class. The Sanders dataset has the most balanced classes and the results show that the SVM achieved the best accuracy for these data. Concerning the false positive rate, most algorithms performed poorly (FPR > 0.3) for this measure. The algorithm with the lowest average FPR was the Naïve Bayes, but its average accuracy was 53.51%. For the SS-Twitter dataset the KNN algorithm achieved the best accuracy and FPR, followed closely by the Naïve Bayes. The decision tree performed very poorly for n-grams with $n > 3$. The best algorithm for SemEval was KNN with 3 neighbors, and the second best algorithm was the Support Vector Machine.

By analyzing each algorithm individually by its average accuracy and FPR, Naïve Bayes achieved an average accuracy of 56.99% and 0.30 FPR for all datasets, SVM achieved an average accuracy of 56.68% and 0.31 FPR. For the J48 the average accuracy was 46.77% while the average FPR was 0.80, and KNN reached an average accuracy of 53.99% and 0.31 FPR. In terms of accuracy, the algorithm with the best overall performance was the Naïve Bayes, which also achieved the lowest average FPR, while the decision tree trained with J48 presented the highest FPR.

In the second set of experiments, the attributes were generated using the dictionary categories. The results are presented in Table 6. For the OMD and Sanders datasets, the Naïve Bayes algorithm presented the best results, with an average accuracy of 51.01% for OMD and 57.21% for Sanders. For the SS-Twitter and SemEval data, the best results were achieved by the SVM algorithm with an average accuracy of 71.63% for SS-Twitter and 69.29% for SemEval. With respect to the most suitable representation of the texts, the LIWC dictionary showed the highest accuracy values. By comparing the results of Tables 4 and 5, it is possible to observe a better accuracy using the standard TF-IDF representation for the OMD, Sanders and SS-Twitter data, and an improvement for the SemEval data when using the category-based representations. It is interesting to observe that for all datasets the

Table 5
Accuracy (ACC), false positive rate (FPR), and F-measure for the documents represented using the TF-IDF approach.

Classifier		OMD			Sanders			SS-Twitter			SemEval		
		ACC	FPR	F-measure	ACC	FPR	F-measure	ACC	FPR	F-measure	ACC	FPR	F-measure
NB	1-gram	56.351	0.397	0.640	55.204	0.243	0.616	34.867	0.787	0.338	57.735	0.301	0.695
	2-gram	63.234	0.078	0.763	55.090	0.190	0.631	75.979	0.075	0.858	46.065	0.638	0.480
	3-gram	**63.011**	**0.060**	**0.766**	52.149	0.155	0.627	**77.102**	**0.041**	**0.868**	47.052	0.633	0.488
	4-gram	42.641	0.064	0.538	52.149	0.110	0.639	77.924	0.021	0.874	46.755	0.636	0.485
	5-gram	40.683	0.033	0.538	53.507	0.095	0.649	78.198	0.011	0.877	46.607	0.639	0.482
	6-gram	40.963	0.061	0.531	51.471	0.098	0.639	70.939	0.179	0.816	46.657	0.638	0.483
SVM	1-gram	49.692	0.206	0.530	60.294	0.357	0.606	49.849	0.523	0.598	56.378	0.334	0.678
	2-gram	47.902	0.204	0.521	62.670	0.379	0.613	61.600	0.344	0.728	60.868	0.231	0.730
	3-gram	47.342	0.163	0.531	**64.253**	**0.374**	**0.625**	58.724	0.377	0.703	60.622	0.240	0.727
	4-gram	52.938	0.441	0.458	64.593	0.355	0.634	58.724	0.377	0.703	60.943	0.235	0.729
	5-gram	53.665	0.425	0.469	64.932	0.333	0.644	67.571	0.236	0.787	60.967	0.235	0.730
	6-gram	55.064	0.614	0.380	64.932	0.319	0.649	71.296	0.159	0.821	60.918	0.235	0.729
J48	1-gram	64.298	0.998	0.003	54.977	0.350	0.578	21.939	0.995	0.009	32.026	0.986	0.027
	2-gram	64.354	0.989	0.022	56.674	0.902	0.176	76.198	0.062	0.861	31.902	0.989	0.023
	3-gram	51.931	0.306	0.507	52.602	0.993	0.014	21.857	0.997	0.007	32.494	0.973	0.051
	4-gram	62.843	0.537	0.470	52.602	0.993	0.014	21.939	0.995	0.009	34.592	0.887	0.193
	5-gram	62.563	0.521	0.477	52.602	0.993	0.014	21.939	0.995	0.009	34.592	0.887	0.193
	6-gram	48.461	0.284	0.498	52.602	0.993	0.014	21.884	0.996	0.008	34.592	0.887	0.193
KNN (3)	1-gram	46.503	0.160	0.528	57.240	0.357	0.588	54.396	0.426	0.663	60.720	0.223	0.731
	2-gram	46.726	0.182	0.523	56.448	0.345	0.588	55.601	0.415	0.674	61.115	0.210	0.737
	3-gram	46.782	0.188	0.521	56.222	0.343	0.588	44.618	0.616	0.521	**61.535**	**0.208**	**0.739**
	4-gram	46.782	0.188	0.521	56.335	0.331	0.593	45.138	0.609	0.527	61.239	0.208	0.738
	5-gram	35.646	0.000	0.526	56.787	0.319	0.600	58.998	0.374	0.705	61.115	0.219	0.735
	6-gram	35.646	0.000	0.526	56.335	0.410	0.562	78.307	0.000	0.878	61.337	0.207	0.738

Table 6
Accuracy (ACC), false positive rate (FPR), and F-measure with attributes based on dictionary category: MRC (MRC Psycholinguistic Database), LIWC (Linguistic Inquiry and Word Count), Stagger (Stanford Pos Tagger) and oNLP (Apache Open Natural Language Processing).

Classifier		OMD			Sanders			SS-Twitter			SemEval		
		ACC	FPR	F-measure	ACC	FPR	F-measure	ACC	FPR	F-measure	ACC	FPR	F-measure
NB	MRC	50.028	0.323	0.491	58.258	0.348	0.598	62.421	0.349	0.731	62.497	0.198	0.746
	LIWC	54.896	0.206	0.557	59.502	0.348	0.605	67.324	0.275	0.777	65.680	0.296	0.738
	sTagger	50.588	0.174	0.544	53.733	0.274	0.599	68.118	0.235	0.790	66.025	0.159	0.773
	oNLP	48.517	0.149	0.541	57.353	0.274	0.618	68.967	0.207	0.800	65.236	0.160	0.769
SVM	MRC	35.478	0.016	0.521	56.335	0.533	0.504	73.459	0.063	0.847	68.838	0.000	0.815
	LIWC	**51.147**	**0.140**	**0.557**	60.068	0.402	0.587	66.749	0.277	0.773	70.762	0.104	0.808
	sTagger	45.048	0.110	0.536	56.787	0.233	0.628	72.638	0.133	0.832	68.715	0.003	0.814
	oNLP	40.963	0.052	0.534	55.204	0.224	0.622	73.678	0.111	0.841	68.838	0.000	0.815
J48	MRC	36.877	0.052	0.517	55.656	0.290	0.603	**77.650**	**0.014**	**0.874**	68.838	0.000	0.815
	LIWC	58.198	0.559	0.429	50.679	0.395	0.538	55.546	0.426	0.669	65.408	0.246	0.750
	sTagger	47.846	0.294	0.491	49.887	0.376	0.542	61.326	0.332	0.730	63.064	0.205	0.748
	oNLP	47.398	0.192	0.523	51.471	0.386	0.546	53.602	0.430	0.658	58.895	0.302	0.700
KNN	MRC	38.165	0.100	0.509	57.466	0.269	0.620	67.078	0.239	0.784	66.223	0.068	0.792
	LIWC	39.787	0.009	0.540	**57.127**	**0.171**	**0.647**	72.364	0.140	0.830	**69.849**	**0.028**	**0.816**
	sTagger	38.500	0.030	0.529	52.828	0.124	0.638	68.803	0.179	0.805	67.111	0.049	0.799
	oNLP	38.444	0.019	0.532	51.471	0.076	0.644	67.872	0.186	0.799	67.209	0.048	0.800

lexicon-based categories resulted in better false positive rates, and F-measure. The results highlighted in Table 5 and Table 6 correspond to the best configuration of classifier and document representation for each dataset.

5. Conclusions and future trends

Polarity classification of Twitter messages is a challenging task for many reasons, including the speed with which messages are generated, the huge number of messages generated around certain subjects, the small length of the messages, and the colloquial nature of the posts. In addition, these messages do not come with an explicit polarity attached. Therefore, to automate polarity classification of tweets, it is necessary to label part of the messages to train a classifier. And this labeling process is very time consuming and subjective, thus prone to error.

To overcome these problems, the present paper introduced a framework that combines a knowledge-based classification with machine learning algorithms to automatically assign polarity to Twitter messages. The knowledge base contains emoticons and sentiment-based words. The process requires finding messages containing emoticons or sentiment-based words with a

minimum coverage of the dataset, selecting these messages for training the classifier, and then labeling the unlabeled messages using a classifier.

The essence of this system is its structure divided into modules, which can be configured according to the application context. The framework offers in the preprocessing module different approaches to structure text messages, and in the classification module various machine learning algorithms can be employed, such as Naive Bayes, SVM, KNN and J48. The experiments performed showed that the process provides a good approach to automatically perform polarity analysis of Twitter messages, with accuracy levels of almost 80%. Future works include the automatic assignment of weights to words and emoticons in order to establish classification priorities. Besides, semi-supervised learning algorithms may be suitable for the classification module, and investigations in this direction will be performed.

Acknowledgments

The authors thank Mackenzie University, MackPesquisa, CNPq, Capes (Proc. n. 9315/13-6) and FAPESP for the financial support.

References

[1] M. Perc, The Matthew effect in empirical data, J. R. Soc. Interface 11 (2014) 1–15.
[2] J. Gao, J. Hu, X. Mao, M. Perc, Culturomics meets random fractal theory: insights into long-range correlations of social and, J. R. Soc. Interface 9 (2012) 1956–1964.
[3] M. Perc, Evolution of the most common English words and phrases over the centuries, J. R. Soc. Interface 12 (2012) 1–6.
[4] J. Park, V. Barash, C. Fink, M. Cha, Emoticon style: interpreting differences in emoticons across cultures, Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM) (2013) 466–475.
[5] M. Tsytsarau, T. Palpanas, survey on mining subjective data on the web, Ingegneria e Scienza dell'Informazione, University of Trento, Trento, Relatório Técnico DISI-10-045, 2010.
[6] Datasift. (2012) Browse data sources – Twitter. [Online]. Available from: http://datasift.com/source/6/twitter (accessed: 12.04.12).
[7] H. Tang, S. Tan, X. Cheng, A survey on sentiment detection of reviews, J. Exp. Syst. Appl. Int. J. Arch. 36 (2009) 10760–10773.
[8] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, Sentiment analysis of Twitter data, Proceedings of the Workshop on Languages in Social Media, 2011, pp. 30–38.
[9] S. Bhuta, A. Doshi, U. Doshi, M. Narvekar, A review of techniques for sentiment analysis of Twitter data, in: Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on, Ghaziabad, India, 2014, pp. 583–591.
[10] P.S. Earle, D.C. Bowden, M. Guy, Twitter earthquake detection: earthquake monitoring in a social world, Ann. Geophys 54 (2011) 708–715.
[11] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, K. Tao, Twitcident: fighting fire with information from social web streams, in: Proceedings of the 21st International Conference Companion on World Wide Web, 2012, pp. 305–308.
[12] M. Yoshida, S. Matsushima, S. Ono, I. Sato, H. Nakagawa, ITC-UT: tweet categorization by query categorization of on-line reputation management, in: Conference on Multilingual and Multimodal Information Access Evaluation, 2010.
[13] A. Tumasjan, T.O. Sprenger, P.G. Sandner, I.M. Welpe, Predicting elections with Twitter: what 140 characters reveal about political sentiment, Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, AAAI, 2010.
[14] A. Bermingham, A. Smeaton, On using Twitter to monitor political sentiment and predict election results, Proceedings of the Sentiment Analysis Where AI Meets Psychology, 2011, pp. 2–10.
[15] J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, J. Comput. Sci. 2 (2011) 1–8.
[16] F. Botta, H.S. Moat, T. Preis, Quantifying crowd size with mobile phone and Twitter data, R. Soc. Open Sci. 2 (2015) 1–6.
[17] C.M. Alis, et al., Quantifying regional differences in the length of Twitter messages, PLoS One 10 (2015) 1–10.
[18] S. Salustiano, O Profissional Analista, Para entender o Monitoramento de Mídias Sociais (2012) 34–40.
[19] Elife. (2012) Serviços. [Online]. Avaliable from: http://elife.com.br/servicos/ (accessed: 30.05.13).
[20] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inform. Retriev. 2 (2008) 1–135.
[21] N.R.S. Filho, Monitoramento das redes sociais como forma de relacionamento com o consumidor. O que as empresas estão fazendo? Gestão Contemp. 1 (2011) 63–86.
[22] H. Yu, V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003, pp. 129–136.
[23] A. Yessenalina, Y. Yue, C. Cardie, Multi-level structured models for document-level sentiment classification, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, 2010, pp. 1046–1056.
[24] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, 2005, pp. 347–354.
[25] A.B. Sayeed, J. Boyd-Graber, B. Rusk, A. Weinberg, Grammatical structures for word-level sentiment detection, in: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Montreal, Canada, 2012, pp. 667–676.
[26] X. Wang, F. Wei, X. Liu, M. Zhou, M. Zhang, Topic sentiment analysis in Twitter: a graph-based hashtag sentiment classification approach, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, Scotland, UK, 2011, pp. 1031–1040.
[27] A. Kumar, T.M. Sebastian, Sentiment analysis: a perspective on its past, present and future, Intell. Syst. Appl. 4 (2012) 1–14.
[28] N.A. Diakopoulos, D.A. Shamma, Characterizing debate performance via aggregated Twitter sentiment, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta, GA, USA, 2010, pp. 1195–1198.
[29] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, A. Kappas, Sentiment strength detection in short informal text, J. Am. Soc. Inform. Sci. Technol. 62 (2010) 2544–2558.
[30] L.M. Santos, Protótipo para mineração de opinião em redes sociais: estudo de casos selecionados usando o Twitter, Universidade Federal de Lavras, Lavras - MG, Monografia (Graduação em Ciência da Computação), 2010.
[31] J. Wiebe, T. Wilson, R. Bruce, M. Bell, M. Martin, Learning subjective language, Comput. Linguist. 30 (2004) 277–308.
[32] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[33] A. Kumar, T.M. Sebastian, Machine learning assisted sentiment analysis, in: Proceedings of International Conference on Computer Science & Engineering, 2012, pp. 123–130.
[34] Y.R. Tausczik, J.W. Pennebaker, The psychological meaning of words: Liwc and computerized text analysis methods, J. Lang. Social Psychol. 29 (2010) 24–54.
[35] A. Esuli, F. Sebastiani, SentiWordNet: a publicly available lexical resource for opinion mining, in: Conference on Language Resources and Evaluation, 2006, pp. 417–422.
[36] M. Annett, G. Kondrak, A comparison of sentiment analysis techniques: polarizing movie blogs, in: Proceedings of the Canadian Society for Computational Studies of Intelligence, 21st Conference on Advances in Artificial Intelligence, 2008, pp. 25–35.
[37] A. Go, R. Bhayani, L. Huang, Twitter Sentiment Classification using Distant Supervision" Technical report, Stanford Digital Library Technologies Project, 2009.
[38] T. Lake, Twitter Sentiment Analysis, Western Michigan University, For client William Fitzgerald, Kalamazoo, MI, 2011.

[39] J. Bollen, H. Mao, A. Pepe, Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena, in: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011, pp. 450–453.
[40] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 10, 2002, pp. 79–86.
[41] R. Prabowo, M. Thelwall, Sentiment analysis: a combined approach, J. Informetrics 3 (2009) 143–157.
[42] J. Read, Using emoticons to reduce dependency in machine learning techniques for sentiment classification, Proceedings of the ACL Student Research Workshop, 2005, pp. 43–48.
[43] M. Araújo, P. Gonçalves, F. Benevenuto, Métodos para análise de sentimentos no Twitter, in: Proceedings of the Simpósio Brasileiro de Sistemas Multimídia e Web (WEBMEDIA), Salvador, 2013.
[44] A. Bakliwal, et al., Mining sentiments from Tweets, in: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, Jeju, Republic of Korea, 2012, pp. 11–18.
[45] N.F.F. da Silva, E.R. Hruschka, E.R. Hruschka Jr., Tweet sentiment analysis with classifier ensembles, Decis. Support Syst. 66 (2014) 170–179.
[46] F.H. Khan, S. Bashir, U. Qamar, TOM: Twitter opinion mining framework using hybrid classification scheme, Decis. Support Syst. 57 (2014) 245–257.
[47] M.M. Bradley and P.J. Lang (2013, January) ANEW Message. [Online]. Available from: http://csea.phhp.ufl.edu/media/anewmessage.html (accessed: 15.09.14).
[48] D. Watson, L.A. Clark, THE PANAS-X Manual for the Positive and Negative Affect Schedule - Expanded Form, The University of Iowa, Iowa, 1994.
[49] E. Cambria, A. Hussain, Sentic Computing: Techniques, Tools, and Applications, Springer, Dordrecht, Netherlands, 2012.
[50] I.H. Witten, Text Mining. In Practical Handbook of Internet Computing, Chapman & Hall/CRC Press, Florida, 2005.
[51] J.W. Pennebaker, M.E. Francis, Linguistic Inquiry and Word Count, Lawrence Erlbaum, 1999.
[52] J.W. Pennebaker, R.J. Booth, M.E. Francis, Linguistic Inquiry and Word Count: LIWC2007 - Operator's Manual, LIWC.net, Austin, Texas, 2007.
[53] K. Toutanova, C.D. Manning, Enriching the knowledge sources used in a maximum entropy part-of-speech Tagger, in: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13, Hong Kong, 2000, pp. 63–70.
[54] E.S. Atwell, J. Hughes, C. Souter, AMALGAM: automatic mapping among lexico-grammatical annotation models, in: Workshop On The Balancing Act: Combining Symbolic And Statistical Approaches To Language, 1994.
[55] C. Fellbaum, WordNet: An Electronic Lexical Database, MIT Press, Cambridge, MA, 1998.
[56] D.A. Shamma, L. Kennedy, E.F. Churchill, Tweet the debates: understanding community annotation of uncollected sources, in: Proceedings of the First SIGMM Workshop on Social Media, Beijing, China, 2009, pp. 3–10.
[57] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed., Morgan Kaufmann, 2011.
[58] I.H. Witten, Text mining. In Practical Handbook of Internet Computing, Chapman & Hall/CRC Press, Florida, 2005.
[59] J. Han, M. Kamber, Jian Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann Publishers, 2011.