Soft Computing (2018) 22:6011–6024
https://doi.org/10.1007/s00500-017-2994-8

FOCUS

Semantic Twitter sentiment analysis based on a fuzzy thesaurus

Heba M. Ismail[1] · Boumediene Belkhouche[1] · Nazar Zaki[1]

Published online: 5 January 2018

Abstract
We define a new, fully automated and domain-independent method for building feature vectors from Twitter text corpus for machine learning sentiment analysis based on a fuzzy thesaurus and sentiment replacement. The proposed method measures the semantic similarity of Tweets with features in the feature space instead of using terms' presence or frequency feature vectors. Thus, we account for the sentiment of the context instead of just counting sentiment words. We use sentiment replacement to reduce the dimensionality of the feature space and a fuzzy thesaurus to incorporate semantics. Experimental results show that sentiment replacement yields up to 35% reduction in the dimensionality of the feature space. Moreover, feature vectors developed based on a fuzzy thesaurus show improvement of sentiment classification performance with multinomial naïve Bayes and support vector machine classifiers with accuracies of 83 and 85%, respectively, on the Stanford testing dataset. Incorporating the fuzzy thesaurus resulted in the best accuracy compared to the baselines with an increase greater than 3%. Comparable results were obtained with a larger dataset, the STS-Gold, indicating the robustness of the proposed method. Furthermore, comparison of results with previous work shows that the proposed method outperforms other methods reported in the literature using the same benchmark data.

Keywords Text mining · Fuzzy thesaurus · Semantic analysis · Text context · Twitter sentiment analysis

1 Introduction

Discourse on social networking sites (SNS) has become predominant in shaping public opinion. A recent meta-review analyzed thirty-six studies on the relationship between the use of social networking sites and everything from civic engagement to tangible actions such as voting and protesting (Boulianne 2015). A key finding of the meta-review indicated that 82% of the factors analyzed in the study showed a positive relationship between the use of SNS and some form of civic or political engagement or participation, especially in the youth populations. This fact emphasizes the importance of sentiment analysis of SNS data in order to understand public opinion on a particular matter. The popularity of Twitter among other social networking sites is noticeably increasing (Liu et al. 2014). Every month millions of people tweet about their likes, dislikes, plans, and other issues. Such a huge amount of public opinions can be valuable in defining strategies. An example of this impact is the US political campaigns (Pew Research Center 2014). Consequently, Twitter sentiment analysis has attracted wide attention in recent years to gauge the public opinion toward specific issues, products, or other targets.

Sentiment analysis of Tweets is a challenging task owing to the highly unstructured nature of the text and its context complexity (Lee and Pang 2008). In addition, the text in a Tweet is condensed into no more than 140 characters, and users can use a countless mixture of formal and informal language, slogans, symbols, emoticons, and special characters to express their opinions conveying different sentiments.

The two most commonly used approaches in sentiment analysis studies are: machine learning, essentially a supervised approach, and score based or lexicon based which is an unsupervised approach (Abbasi et al. 2008). The latter requires comprehensive sentiment lexicons, which are

Communicated by S. Deb, T. Hanne, K.C. Wong.

✉ Heba M. Ismail
hebaismail20@gmail.com

Boumediene Belkhouche
b.belkhouche@uaeu.ac.ae
http://faculty.uaeu.ac.ae/b_belkhouche/

Nazar Zaki
nzaki@uaeu.ac.ae

[1] Department of Computer Science and Software Engineering, College of Information Technology, United Arab Emirates University, Al Ain, UAE

expensive to build, weak in handling context-dependent terms, and are restricted by the static prior sentiment scores that are predefined in the lexicons. Even with some extensions to existing lexicons (Turney and Littman 2003), slogans and jargons commonly used in Twitter are still not accurately analyzed, thereby missing out important sentiment indicators. On the other hand, machine learning approaches rely on vector extraction to represent the most relevant and important text features (Cambria et al. 2013) that can be used to train classifiers such as naïve Bayes (NB) and support vector machines (SVMs). Feature vector extraction methods may eliminate relevant semantic relationships in the text. However, in several cases, the sentiment of a word is implicitly associated with the semantics of its context (Cambria et al. 2013). Several variations of feature vectors have been used to achieve a better realization of context, such as different combinations of *n*-grams (Abbasi et al. 2008; Go et al. 2009; Lima et al. 2015) and parts of speech (POS) features (Lima et al. 2015; Barbosa and Feng 2010). However, these features are used with representations, mostly term frequency or presence, which do not reflect the semantic relationship between words and sentences, or sentences and paragraphs. Recent efforts attempted to explicitly incorporate semantics by means of ontologies (Kontopoulos et al. 2013), or semantic entities (Batra and Rao 2010). Nevertheless, semantic approaches relying on ontologies are limited by their predefined knowledge bases. In addition, methods that rely on sentiment entities are domain dependent and may fail to recognize whether a sentiment is associated with the extracted entity or a different entity. As a result, we need a semantic analysis approach that is domain independent, adaptive to new terms, jargons or slogans, fully automated, and not restricted by previously defined sentiment scores.

In this paper, we extend our previous work (Ismail et al. 2016b) on using a custom fuzzy thesaurus for Twitter sentiment analysis. We define a fully automated method for building semantic Twitter feature vectors for machine learning sentiment analysis based on a fuzzy thesaurus and sentiment replacement. The proposed method measures the semantic similarity of Tweets with features in the feature space instead of simply using occurrences or frequencies. By measuring the semantic similarity, we account for the sentiment of the context instead of just counting sentiment words. This is primarily important in Twitter given the informal writing style that may use positive words to ironically express negative feelings and vice versa. In addition, this method produces less sparse datasets. The fuzzy thesaurus approach for semantic similarity analysis is common in information retrieval on the Web (Ismail 2014; Ogawa et al. 1991; Yerra and Ng 2005) and was shown to result in high accuracy, but it has yet to be used for Twitter sentiment analysis. We also integrate sentiment replacement in our approach to reduce the dimensionality of the feature space. The major contributions of this work are summarized in the following four points:

- Outline a framework for semantic Twitter sentiment analysis based on a fuzzy thesaurus and sentiment replacement.
- Show that using a fuzzy thesaurus can incorporate semantic relationships for Twitter sentiment analysis and increase the accuracy of sentiment analysis.
- Show that using a fuzzy thesaurus to represent semantic relationships yields some improvement over other representations including frequency, presence or polarity, and term frequency inverse document frequency (TF/IDF).
- Show that sentiment replacement can significantly reduce the dimensionality of a Twitter feature space.

In the next section, we will briefly review the literature on the general theme of sentiment analysis. We will highlight some of the analysis issues and challenges associated with unstructured text as found in Twitter. Two major approaches to sentiment analysis are introduced: the lexicon-based approach and the machine learning approaches. Then, we will review some research work related to semantic sentiment analysis. Section 3 presents the fuzzy set information retrieval model that will be adopted in our approach. Section 4 gives a short description of three most commonly used machine learning classifiers to address the Twitter sentiment analysis. These are multinomial naïve Bayes, Bernoulli naïve Bayes, and SVM classifiers. Section 5 presents our approach to Twitter sentiment analysis, including the use of sentiment replacement to reduce the dimensionality of the feature space and the fuzzy thesaurus to incorporate semantics. Section 6 presents the concrete steps we carried out to experiment with our method using the Stanford testing dataset and STS-Gold dataset. In Sect. 7, we will discuss experimental results and compare our method with other methods to demonstrate that our strategies yielded a better performance. Finally, we will summarize our results and highlight potential future research directions in the concluding section.

2 Literature review

Sentiment mining, polarity mining, opinion mining, or sentiment analysis are concerned with the analysis of text containing opinions and emotions (Lee and Pang 2008). Typically, sentiment analysis is carried out on free text or unstructured text, such as online forums (Abbasi et al. 2008), customer reviews (Elfeky and Elhawary 2010), and Twitter (Speriosu et al. 2011). This is because unstructured free text includes more information than structured databases, especially when it concerns unbounded information such as feelings, opinions, and preferences. However, unstructured

texts suffer from several complications. First, unlike structured data, there are no predefined features with known and well-defined values. Unstructured text may contain any number of various words. Second, unstructured text may have the same word used in several ways and in different contexts implying different meanings (polysemous words), or may have many words referring to the same exact meaning (synonymous words) causing redundancy and inconsistencies. Third, in some unstructured text contexts, such as informal social networks like Twitter, it is common to use special characters, emoticons, and abbreviations that add noise to the text and at the same time may add high value if analyzed carefully. These inherent complications in natural language impose greater challenges on text mining tasks such as sentiment analysis. Therefore, data preprocessing is vital in sentiment analysis especially for text collected from social media Web sites because, besides being unstructured, it may contain spelling mistakes and peculiarities. Sentiment analysis tasks include some form of natural language preprocessing, such as tokenization, spellchecking, stopwords removal, and stemming, to produce feature vectors representing the most important text features that can be used later for sentiment classification (Lee and Pang 2008; Abbasi et al. 2008; Go et al. 2009). Moreover, sentiment analysis researchers have come to struggle with NLP's difficult problems, such as coreference resolution, negation handling, anaphora resolution, named-entity recognition, and word-sense disambiguation (Cambria et al. 2013).

In addition to data preprocessing, some of the most important tasks of sentiment analysis where most of the research efforts are focused include class labeling, annotation granularity, and target identification (Jiang et al. 2011). In the class labeling task, some of the research work focuses on categorizing text as subjective or objective. In sentiment analysis, this task is usually carried out first because it has been verified that performing it prior to polarity classification improves the latter (Lee and Pang 2008). In other words, if a text is identified as subjective, then we can perform polarity classification to determine whether this subjective text is carrying positive or negative sentiment. Another active research focus is on alleviating the cost inherent in manual annotation by introducing automatic class labeling, also known as distant supervision. For example, Go et al. (2009) used emoticons such as ":−)" and ":(" to label Tweets as positive or negative. However, Speriosu et al. (2011) argued that using noisy sentiment labels may hinder the performance of sentiment classifiers. They proposed exploiting the Twitter follower graph to improve sentiment classification and constructed a graph that has users, Tweets, word unigrams, word bigrams, hashtags, and emoticons as nodes, which are connected based on the link existing among them (e.g., users are connected to Tweets they created; Tweets are connected to word unigrams that they contain, etc.). They then applied a label propagation method where sentiment labels were propagated from a small set of nodes seeded with some initial label information throughout the graph. Having a preprocessed subjective text with class labels, sentiment polarity classification can be carried out at the document (Zhou and Chaovalit 2005), sentence (Elfeky and Elhawary 2010), or phrase levels (Wilson et al. 2005), where a phrase is part of a sentence, which we refer to as the granularity of the classification. Finally, recognizing the source and the target of a sentiment is considered as one of the challenges of sentiment analysis that was addressed by number of researchers (Perez-Tellez et al. 2010).

The three main approaches used in sentiment analysis studies are machine learning essentially a supervised approach, link analysis, and score based or lexicon based which is an unsupervised approach (Abbasi et al. 2008). We will focus on explaining only the two most commonly used approaches in sentiment analysis, the lexicon-based approach and the machine learning approach. In the machine learning approach, we will explore the most commonly used machine learning classifiers, namely multinomial naïve Bayes, Bernoulli naïve Bayes, and support vector machines, to address the Twitter sentiment analysis problem in the literature (Go et al. 2009; Speriosu et al. 2011; Pang et al. 2002; Saif et al. 2012).

2.1 Lexicon-based approaches

Lexicon-based approaches are widely used to classify unsupervised text sentiment. Such techniques attempt to classify data based on the number of positive, and negative words present in the text and do not need any training dataset. These positive or negative words, which express opinions and emotions, are known as "opinion words" or "affect terms," and the lexicon is known as "opinion lexicon" or "affect lexicon." In the lexicon-based approaches, researchers rely on external lexical resources that associate a polarity score to each term. The sentiment of a text depends on the overall sentiment score of all the terms that compose it. Examples of opinion and emotion lexicons are SentiWordNet (Esuli 2006), WordNetAffect (Strapparava and Valitutti 2004), and SenticNet (Cambria et al. 2010). Although relying on predefined lexicons eliminates the need for training data and mostly allows for domain-independent sentiment analysis, there exist a number of significant disadvantages associated with this approach. First, there is no mechanism to deal with context-dependent words. For example, the word "Long" can be used to convey a positive as well as a negative opinion both depending on the context in which it is used. We can think of two sentences such as "This mobile takes long time to charge," which is a negative opinion, and "This mobile phone has a long battery life," which implies a positive opinion. Another common example is the use of negated words. For example, "This car is not good at

all" implies an extreme negative sentiment, but by calculating the scores of sentiment words it will have a high positive score since it contains the word "good," which has an extremely positive score in any lexicon. Second, the process of building sentiment lexicons is costly and often requires manual annotation of terms, which is sometimes subjective. To reduce the cost associated with building the lexicon, some researchers use automatic approaches based on association in which the score for each new polarity term is calculated using the frequency of the proximity of that term with respect to one or more seed words (Taboada et al. 2011). Seed words represent a small set of words with strong negative or positive associations such as excellent or abysmal. Third, lexicon-based methods are restricted by the static prior sentiment scores that are predefined in the lexicons. Therefore, sentences containing sentiment terms that are not defined in the lexicon will not be accurately scored. To avoid some of these drawbacks, researchers use learning approaches (Abbasi et al. 2008; Go et al. 2009; Lima et al. 2015).

2.2 Machine learning approaches

Machine learning approaches to opinion mining rely on vector extraction to represent the most relevant and important text features (Cambria et al. 2013) that can be used to train classifiers such as naïve Bayes (NB) and support vector machines (SVMs). The most commonly used features are *n*-grams—typically unigrams and bigrams (Abbasi et al. 2008; Go et al. 2009; Lima et al. 2015). Moreover, natural language processing is sometimes used to extract parts of speech information (for example, nouns, adjectives, adverbs, and verbs) as a basic form of word-sense disambiguation (Barbosa and Feng 2010; Lima et al. 2015). The two most commonly used feature vector representations are term frequency and presence (Ismail et al. 2016a). Frequency indicates the number of times a feature is present in an instance. Presence is a binary-valued feature vector in which the entries indicate only whether a term occurs (value 1) or does not (value 0). It is also possible to use other term-based features representations. For example, Lima et al. (2015) used term frequency inverse document frequency (TF/IDF). Although these techniques partially alleviated some challenges inherent in mining unstructured texts, other additional major challenges surfaced. For instance, converting unstructured text into a feature vector with a term weight assigned as a value helped in imposing a structure into the free text, which in turn made text mining tasks possible, but eliminated the context in terms of the relationship between words in a sentence and between sentences in a paragraph. Yet, in many cases, the sentiment of a word is implicitly associated with the semantics of its context (Cambria et al. 2013).

2.3 Semantic sentiment analysis

Various approaches were proposed in the literature to account for semantic sentiment. Saif et al. (2016) classified these approaches into two categories: contextual semantic approaches and conceptual semantic approaches. In contextual approaches, statistical analysis of the relationships between terms in the text is analyzed. These relationships are mainly co-occurrences. For example, Turney and Littman (2003) used pointwise mutual information (PMI) to measure the statistical correlation between a given word and a balanced set of 14 positive and negative paradigm words (e.g., good, nice, nasty, poor). A word has positive orientation if it has a stronger degree of association to positive words than to negative words, and vice versa. These approaches are limited by the choice of seed words that have predefined extreme polarity. As a result, words that assume positive as well as negative sentiments based on the context are not accurately classified. Consider the word "long" in these two sentences: "my mobile battery has a long life" + and "I had to wait in a long queue before collecting my tickets" −. In these two cases, will the word "long" be associated with a positive sentiment or a negative sentiment? On the other hand, conceptual approaches of semantic sentiment analysis rely on external semantic knowledge bases such as ontologies and semantic networks. For example, Kontopoulos et al. (2013) proposed a framework for semantic sentiment analysis based on the concepts and properties included in the ontology. Although conceptual semantic sentiment might be more comprehensive in terms of concepts diversity along with their semantic relevance, it is still limited by their underlying knowledge bases. Another limitation particularly related to Twitter sentiment analysis is that it is full of jargon and special characters which are usually not considered in knowledge bases, but may indicate important sentiments. More recently, some efforts were directed toward the analysis of entity sentiment rather than term sentiment with the assumption that some entities are more often associated with either positive or negative sentiment. Batra and Rao (2010) proposed a framework for sentiment analysis based on probabilistic models that measure the sentiment of an entity as an aggregation of the sentiment of all Tweets that are associated with that entity. However, there are two limitations associated with this approach. First, automatic entity extraction algorithms rely on predefined manually created lexicons that are most of the time domain dependent resulting in less effective entity extraction for texts belonging to other domains. Second, sometimes the sentiment of the text is actually not associated with the extracted entity resulting in inaccuracies. For example, "The new Twitter for iPhone is awesome." expresses a positive sentiment toward "Twitter," but not toward "iPhone" (Saif et al. 2016).

To overcome the above-mentioned limitations associated with semantic sentiment analysis, we propose a new approach based on a fuzzy thesaurus. Fuzzy thesauri have been widely used in information retrieval and have proved effective in various contexts (Ismail 2014; Ogawa et al. 1991; Yerra and Ng 2005). The major strengths of these thesauri are that they are not limited to specific domains or predefined seed terms or entities and do not require any manual annotation, and they are built and used automatically. Fuzzy thesauri are built based on concepts from fuzzy set information retrieval models (Kraft et al. 1999).

3 Fuzzy set information retrieval model

The fuzzy set theory relies on two main principles: sets are not crisp (boundaries of the sets are ambiguous or fuzzy), and elements belong to the fuzzy set at different levels of membership (Zadeh 1965). Language sentences and documents are typical examples of fuzzy sets. A fuzzy set IR model is adopted to determine the degree of membership between every keyword in a sentence and a fuzzy set that contains different words, each of which belongs to the set at some degree of membership. The degrees of similarity or membership, also referred to as the correlation factors among words, are given by a function which assigns a value in the range [0, 1] to any two words. Hence, if two sentences contain many terms that belong to the same fuzzy sets at a high degree of membership, then the two sentences are similar. There are several methods to define the correlation factors among different words; for example (i) word connection calculates the correlation of any two words w_1 and w_2 by counting the number of documents in a collection C where both w_1 and w_2 appear together, (ii) keyword co-occurrence not only considers the number of documents in a collection where both words w_1 and w_2 appear together, but it also considers the frequency of co-occurrence of both w_1 and w_2 in a document, and (iii) distance considers the frequency of occurrence as well as the distance, which is measured by the number of words, between w_1 and w_2 within a document (Garcia and Ng 2006).

Ogawa et al. (1991) adopted a fuzzy set IR model to determine whether a keyword in a sentence belongs to a fuzzy set that contains words with different levels of similarities among them. They called the fuzzy set a keyword-connection-matrix and defined it as a type of thesaurus that describes relations between keywords by assigning similarity grades restricted to the interval [0, 1]. Yerra and Ng (2005) used the same keyword-connection-matrix proposed by Ogawa et al. (1991) to detect similar HTML documents. Using the keyword-connection-matrix, Yerra et al. compared every keyword, k, in a sentence, i, with every keyword, w, in a document, d, and calculated a word–sentence similarity, $\mu_{k,d}$, using the following fuzzy association:

$$\mu_{k,d} = 1 - \Pi\,(1 - Cf_{kw}) \tag{1a}$$

where Cf_{kw} is the fuzzy relationship between k and w. The average of all μ-values is calculated to yield the overall similarity, *Sim(i,d)*, between i and d as follows:

$$Sim\ (i, d) = \left(\mu_{k1,d} + \mu_{k2,d} + \cdots + \mu_{kn,d}\right)/n \tag{1b}$$

Later, Ismail (2014) adopted the fuzzy set information retrieval model to build a custom fuzzy thesaurus that is used to search for relevant Really Simple Syndication (RSS) feeds based on some user-built concept maps. The custom fuzzy thesaurus built for a specific concept map proved to be very effective in retrieving relevant documents. A similar approach will be adopted in this research, and this will be further explained in subsequent sections.

4 Machine learning classification techniques

Multinomial naïve Bayes, Bernoulli naïve Bayes, and SVM classifiers are the most commonly used machine learning classifiers to address Twitter sentiment analysis (Lee and Pang 2008; Go et al. 2009; Speriosu et al. 2011; Pang et al. 2002).

4.1 Multinomial naïve Bayes text classifiers

A multinomial naïve Bayes (MNB) classifier is a probabilistic classification model based on the Bayes theory. Using the MNB text classifier, the probability of a document, $\boldsymbol{d}$, being in class, $\boldsymbol{c}$, is computed as (Manning et al. 2009):

$$P\,(c|d) \propto P\,(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

where $P(t_k|c)$ is the conditional probability of the term, t_k, occurring in a document of class, c. We interpret $P(t_k|c)$ as a measure of how much evidence t_k contributes in determining that $c$is the correct class. $P(c)$ is the prior probability of a document occurring in class, c. If the terms of a document do not provide clear evidence for one class versus another, we choose the one that has a higher prior probability. $(t_1, t_2, \ldots, t_{nd})$ are the tokens in d that are part of the vocabulary we use for classification, and *nd* is the number of such tokens in d. For example, $(t_1, t_2, \ldots, t_{nd})$ for the one-sentence document "*Beijing and Taipei join the WTO*" might be (Beijing, Taipei, join, WTO), with $n_d = 4$, with the term "*and*" treated as a stopword. In text classification, our goal is to find the *best* class for the document. The best class in M

NB classification is the most likely or *maximum posteriori* (MAP) class, c_{map}:

$$c_{map} = \arg\max_{c \in C} \hat{P}\ (c|d) = \arg\max_{c \in C} \hat{P}\ (c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

4.2 Bernoulli naïve Bayes text classifiers

An alternative to the multinomial model is the multivariate Bernoulli model or Bernoulli model, which generates an indicator for each term of the vocabulary, either 1 indicating the presence of the term in the document or 0 indicating absence.

Usually multinomial naïve Bayes is used when the multiple occurrences of the words matter a lot in the classification problem. Alternatively, the Bernoulli naïve Bayes can be used when the absence of a particular word matters.

4.3 Support vector machines classifiers

The support vector machine (SVM) is a non-probabilistic binary linear classifier that constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. The main underlying idea behind SVM in sentiment classification is to find a hyper plane that divides the documents, or in our case, Tweets as per the sentiment, and the margin between the classes being as high as possible (Bhuta et al. 2014).

To discriminate between "positive" and "negative" Tweets as per the sentiment, the SVM learns a classification function from a set of positive examples $\chi+$ and set of negative examples $\chi-$. According to (Zaki et al. 2009), the classification function takes the form:

$$f\ (x) = \sum\nolimits_{i:x_i \in \chi_+} \lambda_i K\ (x, x_i) - \sum\nolimits_{i:x_i \in \chi_-} \lambda_i K\ (x, x_i)$$

where the nonnegative weights λ_i are computed during training by maximizing a quadratic objective function and the kernel function K(.,.). Any new Tweet, x, is then predicted to be positive or negative if the function, f(x), is positive or negative, respectively. Details about the theory of SVM and how the weights, λ_i, are computed can be found in (Vapnik and Vapnik 1998).

5 Methodology

In our research, we aim to provide a new method for generating semantic feature vectors with reduced dimensionality for Twitter sentiment classification from raw Twitter data. Twitter data can be collected using the Twitter API (https://dev.twitter.com/rest/public), or can be benchmark data, which is publicly available for experiments and research, such as the datasets we use in Sect. 6. We used sentiment replacement to reduce the dimensionality of the feature space as well as a fuzzy thesaurus to incorporate semantics. Our method comprises the following three main tasks, highlighted with a gray rectangle in Fig. 1:

1. Sentiment replacement;
2. Feature extraction and reduction; and
3. Feature vectors generation based on semantic similarities.

The generated semantic feature vectors are then used to train any machine learning classifier for sentiment classification task. We will subsequently present the classification results of MNB, BNB and SVM classifiers in Sect. 6. In the following sections, we will describe the three main tasks of our method.

5.1 Sentiment replacement

Sentiment replacement was carried out programmatically by interfacing with a publicly available Twitter slogan, special characters, emoticons, and abbreviation list. In available sentiment lexicons, only proper and formal words are considered. However, in social networks the use of slogans, emoticons, and abbreviations is very common and it adds strong indication of the sentiment of the text. Such abbreviations and slogans might be removed through natural language processing stages during preprocessing, especially special characters and emoticons, thereby cutting out useful sentiment indicators. Thus, we carried out sentiment replacement of slogans and abbreviations before the preprocessing phase. For example, when "loool" is encountered, we replace it with "Happy." All emoticons are replaced with their equivalent sentiment word. For example, "☺" is replaced with "Happy" and "☹" ":/", ": \" are replaced with "Sad."

5.2 Feature extraction and reduction

Once we have completed sentiment replacement, the natural language processing (Kao and Poteet 2007) of Twitter data was carried out. Generally, unstructured texts cannot be directly processed by classifiers and learning algorithms. In addition, Twitter data are full of peculiarities owing to the informal writing style commonly used on Twitter resulting in more noisy text. Thus, we carried out a number of natural language processing tasks that have proved effective in previous studies (Lee and Pang 2008; Abbasi et al. 2008; Go et al. 2009) and have become a common practice in Twitter preprocessing for sentiment classification to transform the Twitter unstructured text into a 'bag-of-words" model with a reduced number of features, which is manageable by

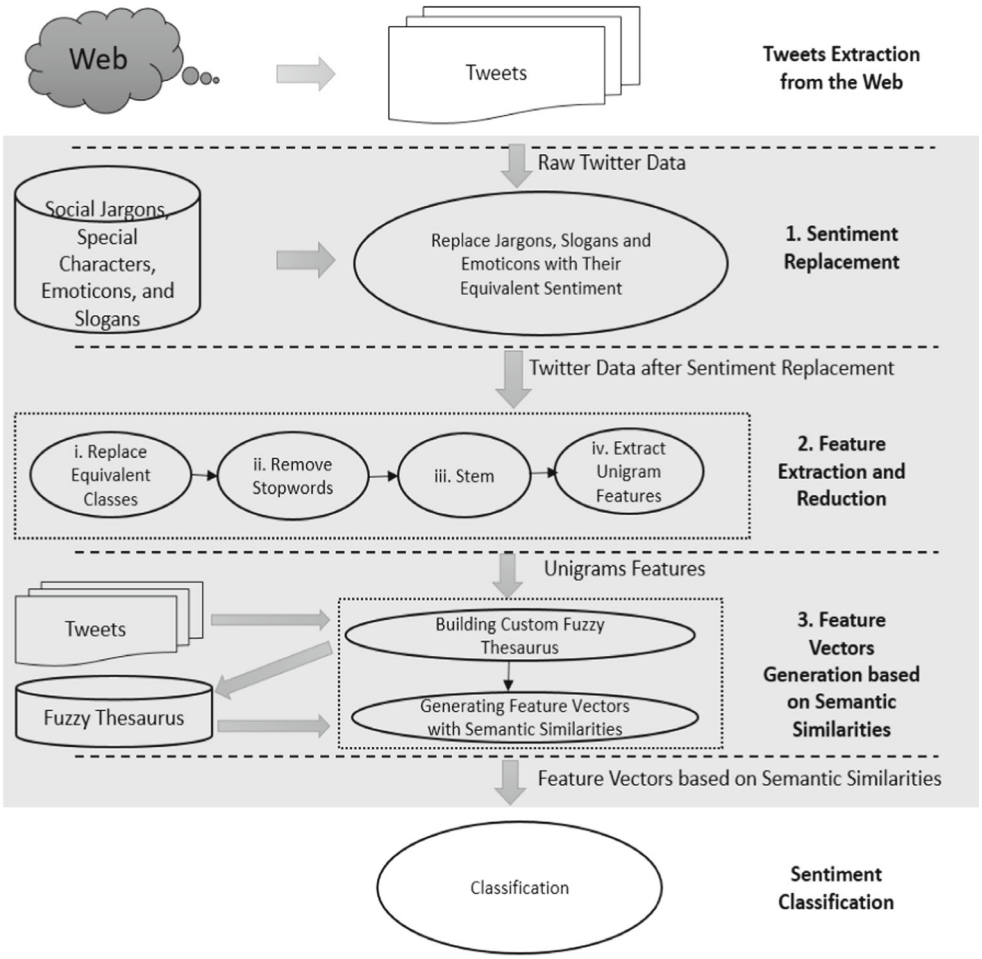

Fig. 1 Semantic sentiment classification based on sentiment replacement and fuzzy thesaurus

classification algorithms. We carried out the following preprocessing tasks in the order described below:

1. Equivalence classes replacement such that:
   - All Twitter usernames that start with @ symbol are replaced with the term "USERNAME";
   - All URL links in the corpus are replaced with the term "URL";
   - The number of letters that are repeated more than twice in all words are reduced; for example, the word "looooooveeee" becomes "loovee" after reduction;
   - All Twitter hashtags that start with the symbol "#" are removed;
2. Stopwords removal: stopwords usually refer to the most common words in a language and are considered to have little meaning; for example, in English some stopwords are: "a," "an," "and," "are," "as," "at," "be," "but," "by";
3. Stemming (Porter 1980) is a process of eliminating the most common morphological and inflectional endings from words in a language with the assumption that all words derived from the same stem share the same meaning; and
4. Bag-of-words extraction, in which we choose unigram features since they can be directly used with the fuzzy association rule as in Eq. (5), typically in a unigram representation, each single word in the corpus is treated as a feature.

After completing the preprocessing tasks, a custom fuzzy thesaurus is built and is used to generate feature vectors based on semantic similarities, which is later used for sentiment classification. The process of building the custom fuzzy thesaurus and generating the feature vectors based on semantic similarities is explained in the following section.

5.3 Feature vectors generation based on semantic similarities

Once features are extracted from the Twitter corpus, the fuzzy thesaurus is built and is used to generate the semantic feature vectors using Eq. (5), adapted from the fuzzy association rule (1a).

5.3.1 Building the custom fuzzy thesaurus

We built the custom fuzzy thesaurus that defines the semantic similarity between each two distinct words in the Twitter corpus by calculating the distance correlation factors between each two distinct words in the corpus using Eqs. (2), (3), and (4). We selected the distance correlation factor since it has been empirically shown to achieve the best results in the information retrieval context with an accuracy rate of 94% compared to 47% for the keyword-connection factor and 52% for the co-occurrence factor (Garcia and Ng 2006). This is because the distance correlation factors account for the frequency and co-occurrence at the same time.

Unigrams features generated in the previous step are now used to generate vectors of all distinct preprocessed words in the Twitter corpus along with the documents' IDs in which they appear, a Tweet is considered a document in this context, and their positions in every document. For example, "mobile" after preprocessing becomes "mobil," and "computer" after preprocessing becomes "comput"; vectors' entries for the words "mobil" and "comput" would be as illustrated in Tables 1 and 2.

Using the vectors of distinct words in the Twitter corpus, we define for every pair of keywords across all documents, the frequency of co-occurrence and relative distance in a single document (C_{ij}), Eq. (2), the normalized value ($\mathrm{nC_{ij}}$), Eq. (3), and finally the distance correlation factor (Cf_{ij}), Eq. (4). These are computed as follows:

Table 1 Vector entry for the word "Mobil"

Term	Document ID	Position
Mobil	Doc1	1, 8
	Doc2	12, 30
	Doc3	7, 10, 27
	Doc4	30

Table 2 Vector entry for the word "Comput"

Term	Document ID	Position
Comput	Doc1	5,30
	Doc5	17, 38
	Doc6	10, 29

$$C_{i,j} = \sum_{x \in V(wi)} \sum_{y \in V(wj)} \frac{1}{d(x,y)} \tag{2}$$

$$nC_{i,j} = \frac{Ci,j}{|V(wi)| \times |V(wj)|} \tag{3}$$

$$Cf_{i,j} = \frac{\sum_{m=1}^{k} nC_{i,j}}{k} \tag{4}$$

where $d(x,y) = |Position(x) - Position(y)| + 1$ is the distance or the number of words between word x and y in a single Tweet, where x is an element of $V\ (wi)$ and y is an element of $V\ (wj)$. $V\ (wi)$ and $V\ (wj)$ are the sets of all occurrences of words *wi & wj* in a single Tweet. To calculate the frequency of co-occurrence and relative distance in a single document, we sum up the inverse distance of every two occurrences of *wi and wj* in that common document. For example, the words "mobil" and "comput" appear together in Doc1; hence, $V(mobil) = \{1, 8\}$, $V(comput) = \{5, 30\}$, and $C_{mobil,comput} = (1/d(1,5) + 1/d(1,30) + 1/d(8,5) + 1/d(8,30))$. If they appear together in other documents, then we have to repeat the same calculation for every common document.

$|V(wi)|\ and\ |V(wj)|$ represent the number of words in $V\ (wi)$ and $V\ (wj)$, respectively, i.e., the frequency of wi and wj in a common Tweet. For example, $|V(mobil)| = |V(comput)| = 2$ in Doc1. Hence, to calculate the normalized frequency of co-occurrence and relative distance for "mobil" and "comput" in Doc1, we compute $\mathrm{nC}_{mobil,comput} = C_{mobil,comput}/(2^*2)$.

The index, m, is in the range $1 \leq m \leq k$ and represents the m_{th} Tweet out of the k Tweets in which both wi and wj occur together. For the words "mobil" and "comput," $m = k = 1$. By dividing the sum of normalized values by the number of common documents between every two words in the corpus, distance correlation factors, Cf, are calculated relevant to the size of the corpus. As a result, a matrix of all distinct words and their semantic relationships is constructed. This matrix is the custom fuzzy thesaurus which is used to measure the partial similarity and exact match between attributes in the feature space and single terms in each single Tweet.

5.3.2 Generating feature vectors with semantic similarities

Once the fuzzy thesaurus is constructed, every feature, F_i, is compared with every word, W_j, in a Tweet, d, thus retrieving the corresponding distance correlation factor, Cf_{ij}, from the custom fuzzy thesaurus, which indicates the word–word semantic similarity.

Once a feature, F_i, is compared to each word, W_j, in a given Tweet, d, the semantic similarity between the feature and the entire Tweet is calculated using Eq. (5), which indicates the word–sentence semantic similarity. This is done for each feature in the feature space against each single Tweet in the corpus as illustrated in Fig. 2.

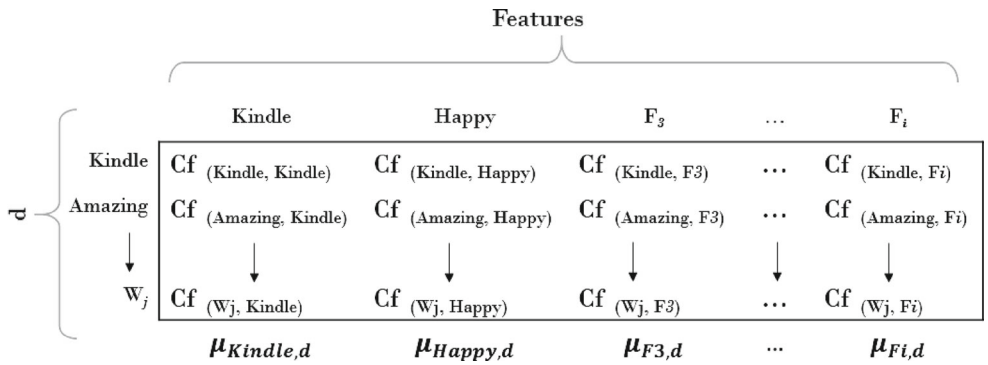

Fig. 2 Calculating the word–sentence semantic similarity ($\mu_{F,d}$) between each feature (F_i) in the feature space and each Tweet (d) in the Twitter corpus

Table 3 Statistics of the Twitter datasets used in this paper

Dataset	Number of Tweets	Positive	Negative	Type
STS-Gold Tweet	2032	632	1400	tenfold cross-validation
Stanford Twitter Sentiment (STS) testing set	359	182	177	tenfold cross-validation

Using this approach, we account for the semantic relationship between each feature with each single Tweet in the corpus allowing for analyzing the overall context instead of just considering the occurrence or the frequency of features in each Tweet.

$$\mu_{F,d} = 1 - \Pi \left(1 - Cf_{ij}\right) \tag{5}$$

6 Experimental work

In this section, we introduce the benchmark datasets that we used in our experiments, our baselines, sentiment replacement and preprocessing, classification based on a fuzzy thesaurus, and finally evaluation measures.

6.1 Dataset

We used the STS-Gold Tweet[1] dataset and the Stanford Twitter Sentiment (STS)[2,3] testing dataset to evaluate the effectiveness of the proposed method (see Table 3 for details). The STS-Gold Tweet dataset contains 2032 randomly collected Tweets, which were manually annotated into positive and negative by three annotators. All the annotators agree on the sentiment of the Tweets in the dataset. The Stanford Twitter Sentiment testing set consists of 359 Tweets that were collected by searching the Twitter API with specific queries including products names, companies, and people and were also manually annotated into positive and negative. We did not use the original Stanford training dataset because it was automatically annotated using emoticons. Although automatic sentiment annotation of Tweets using emoticons is fast, its accuracy is arguable because emoticons might not reflect the actual sentiment of Tweets (Saif et al. 2013). Another limitation of the Stanford original training set is that the set was automatically annotated based on emoticons, but then the emoticons were removed; hence, if we train a classifier on the Stanford training dataset, it will not recognize the emoticons that were initially used for class labeling. Therefore, in this study, we only considered the STS testing dataset and STS-Gold Tweet dataset applying a tenfold cross-validation to both.

6.2 Baselines

We compared the performance of our approach using the fuzzy thesaurus and sentiment replacement with the baselines described below. Although word unigrams are the simplest features being used for sentiment analysis of Tweets, there is evidence that using n-gram features might hinder the accuracy of Twitter sentiment analysis owing to the large number of infrequent words and that unigrams produce better accuracy results (Barbosa and Feng 2010; Ismail et al. 2016a).

[1] STS-Gold dataset can be requested from the authors at: http://kmi.open.ac.uk/people/member/hassan-saif

[2] Stanford dataset official page: http://help.sentiment140.com/for-students

[3] Stanford testing and training datasets can be downloaded from: https://docs.google.com/file/d/0B04GJPshIjmPRnZManQwWEdTZjg/edit

In addition, models trained from word unigrams were shown to outperform random classifiers by a decent margin of 20% (Agarwal et al. 2011); hence, we opted to use only unigram features. We did not perform sentiment replacement for the baselines.

6.2.1 First baseline: unigrams features with polarity

We used the NB classifiers and the SVM classifier trained from word unigrams on polarity dataset as our first baseline model. Polarity indicates whether a feature occurs or not in a Tweet.

6.2.2 Second baseline: unigrams features with frequencies

We used the NB classifiers and the SVM classifier trained from word unigrams on frequency dataset as our second baseline model. Frequencies indicate how many times a feature occurs in a Tweet.

6.2.3 Third Baseline: unigrams features with TF/IDF

We used the NB classifiers and the SVM classifier trained from word unigrams on a term frequency inverse document frequency (TF/IDF) dataset as our third baseline model. TF/IDF is a measure that is intended to reflect how important a word is to a document in a collection or corpus. We calculate TF/IDF as follows:

- *TF(t,d) = Term frequency(t,d)* is the number of times that term, t, occurs in a document, d.
- *IDF(t,D) = Inverse term frequency(t,D)* measures the importance of term, t, in all documents by dividing the total number of documents, N, by the number of documents containing the term, DF, and then taking the logarithm of that quotient.

$$\mathrm{IDF(t,D)} = \log_2 (\mathrm{N/DF})$$

- Finally, the weight is obtained by multiplying the two measures:

$$\mathrm{TF\text{-}IDF(t,d)} = \mathrm{TF(t,d)} * \mathrm{IDF(t,D)}$$

6.3 Sentiment replacement, preprocessing, and feature reduction

Initially, all slogans and abbreviations that have sentiment meaning were searched in the raw Twitter corpus and were replaced with their sentiment equivalence according to the slogan list available in (Just English 2014). Once we have carried out the sentiment replacement, the natural language processing (Kao and Poteet 2007) of Twitter data was carried out. In Table 4, we provide the list of APIs and techniques that we used for preprocessing and feature extraction.

Table 4 APIs and techniques applied for NLP

Natural language processing task	API/technique
Stopwords removal	Apache Lucene core 5.3.0[a]
Stemming	Porter stemming algorithm[b]
Unigram extraction	Apache Lucene core 5.3.0
Equivalence class replacement	Java Regex[c]

[a] https://lucene.apache.org/core/
[b] https://tartarus.org/martin/PorterStemmer/java.txt
[c] More about regex can be found at: https://docs.oracle.com/javase/tutorial/essential/regex/

Table 5 Effect of preprocessing and feature reduction in the feature space size of the Stanford testing dataset

Preprocessing/feature reduction	Feature space size	% of reduction
None	2455	0
Sentiment replacement of slogans, abbreviations, and emoticons	1593	35.11
User names	1605	34.62
URL	1614	34.26
Hashtags	1678	31.65
Repeated letters	1682	31.49
All	1442	41.26

To illustrate the impact of sentiment replacement on reducing feature space dimensionality, we summarized in Table 5 the effect of preprocessing and feature reduction in reducing dimensionality of the original feature space of the Stanford testing dataset. After completing all the preprocessing steps explained in Sect. 5.2, the feature space size was reduced by 41.26%. The most significant contributor to the feature space dimensionality reduction is the sentiment replacement of slogans, abbreviations, and emoticons. The same steps were applied to the STS-Gold dataset.

6.4 Sentiment classification

We developed a Java program using JDK 8 and JRE 8 to build the fuzzy thesaurus and generate the semantic feature vectors (SFV) from a Twitter corpus. Table 6 shows the algorithm for generating semantic feature vectors (SFV). The algorithm expects the following as inputs:

1. Twitter data consisting primarily of messages and sentiment class. Additional data can be present such as user

Table 6 Generating feature vectors based on semantic similarities

Algorithm 1 Generating Feature Vectors based on Semantic Similarities

for each word $W_j \in T$ **do**
  if $W_j \in ASEL$ **then,**
    replace W_j *with equivalent sentiment*
  end if
end for
for each Tweet $d = (d_1, d_2, \ldots, d_n) \in T$ **do**
  replace equivalent classes
  remove stopwords
  perform stemming
  generate word-document-position vectors WDPV
  extract unigrams features F
end for
for each word $W_j \in$ WDPV **do**
  for each word $W_i \in$ WDPV **do**
    if W_j & W_i appear in the same document **then**,
      calculate the frequency of co-occurrence and relative distance in a single document *Cij*
      calculate the normalized value *nCij*
      calculate the distance correlation factor *Cfij*
    end if
  end for
end for
for each Tweet $d = (d_1, d_2, \ldots, d_n) \in T$ **do**
  for each Feature $F = (F_1, F_2, \ldots, F_i) \in F$ **do**
    calculate Feature-Tweet semantic similarity $\mu_{Fi,dn}$
  end for
end for
return semantic feature vectors SFV

ID, hashtags, and queries, which will be preprocessed during natural language processing phases.
2. List of slogans, abbreviations, and emoticons with their corresponding sentiment meaning.

In our implementation on a 2.6 GHz PC running Windows 10, we used the following data representations for inputs:

1. T: Twitter data represented in a list of string arrays, where, each node, d, holds a single record from the Twitter data composed of Strings' elements, for instance Tweet messages, user ID … etc.
2. *ASEL*: slogans, abbreviations, and emoticons represented in a string array. We used the ASEL available in (LOL, OMG and ILY: 60 of The Dominating Abbreviations 2014).

In the intermediate steps, features, F, are represented using a list of strings, the fuzzy thesaurus, comprising all *Cf* values, is represented using a hash table, and word–document–position vectors, WDPV, illustrated in Tables 1 and 2, are represented using user-defined data types. As output, the algorithm returns semantic feature vectors (SFV) and exports them to a comma-separated file ready for classification.

Table 7 Stanford testing set—all features

Unigrams-1442 Features		BNB	SVM	MNB
Polarity-based baseline	Accuracy	76.60%	74.37%	79.38%
	Recall	0.766	0.744	0.794
	Precision	0.766	0.744	0.795
Frequency-based baseline	Accuracy	74.37%	71.86%	79.94%
	Recall	0.744	0.719	0.799
	Precision	0.745	0.719	0.8
TF/IDF-based baseline	Accuracy	76.88%	77.99%	81.89%
	Recall	0.769	0.780	0.819
	Precision	0.769	0.780	0.819
Semantic feature vectors-SFV	Accuracy	71.87%	74.65%	80.78%
	Recall	0.719	0.747	0.808
	Precision	0.719	0.747	0.809

Subsequently, we used Weka 3.8 (Witten et al. 2016) to train the classification model and tested it with a ten-fold cross-validation. Tables 7, 8, 9, and 10 present the classification results of Bernoulli naïve Bayes (BNB), multinomial naïve Bayes (MNB) and SVM classifiers trained on

Table 8 Stanford testing set—selected features

Unigrams-selected features using (IG)		BNB	SVM	MNB
Polarity-based baseline	Accuracy	80.2%	81%	81.62%
	Recall	0.802	0.811	0.816
	Precision	0.844	0.855	0.855
Frequency-based baseline	Accuracy	77.15%	79.10%	82.17%
	Recall	0.772	0.791	0.822
	Precision	0.785	0.828	0.851
TF/IDF-based baseline	Accuracy	80.78%	81.89%	81.62%
	Recall	0.808	0.819	0.816
	Precision	0.842	0.846	0.850
Semantic feature vectors—SFV	Accuracy	77.99%	84.96%	83.29%
	Recall	0.78	0.85	0.833
	Precision	0.808	0.869	0.856

Table 9 STS-Gold—all features

Unigrams-3850 Features		BNB	SVM	MNB
Polarity-based baseline	Accuracy	75.78%	80.17%	81.1%
	Recall	0.758	0.802	0.811
	Precision	0.75	0.796	0.807
Frequency-based baseline	Accuracy	74.60%	81.25%	80.70%
	Recall	0.746	0.813	0.807
	Precision	0.747	0.808	0.806
TF/IDF-based baseline	Accuracy	64.03%	79.33%	77.41%
	Recall	0.640	0.793	0.774
	Precision	0.729	0.787	0.768
Semantic feature vectors—SFV	Accuracy	73.75%	80.5%	80.44%
	Recall	0.737	0.805	0.804
	Precision	0.774	0.804	0.808

unigrams with polarities, frequencies, TF/IDF, and semantic feature vectors (SFV) using a tenfold cross-validation before and after applying an *Information Gain (IG)* attribute selection filter. For sentiment mining, this size of corpus may not provide sufficient coverage of representative sentiment terms and contexts. Therefore, we opted to apply attribute selection filter to eliminate the effect of sentimentally insignificant attributes. *Information Gain (IG)* is used to select subsets of features that are highly correlated with the class while having low inter-correlation. In other words, we selected the features with the highest information gain and removed features with very low information gain from the feature space (Hotho et al. 2005).

Table 10 STS-Gold—selected features

Unigrams—selected features using (IG)		BNB	SVM	MNB
Polarity-based baseline	Accuracy	75.29%	79.87%	80.56%
	Recall	0.753	0.799	0.806
	Precision	0.74	0.796	0.813
Frequency-based baseline	Accuracy	77.21%	81.5 %	82.03%
	Recall	0.772	0.815	0.820
	Precision	0.763	0.815	0.823
TF/IDF-based baseline	Accuracy	79.23%	77.36%	75.49%
	Recall	0.792	0.774	0.755
	Precision	0.785	0.780	0.804
Semantic feature vectors—SFV	Accuracy	80.54%	81 %	82.17%
	Recall	0.805	0.809	0.822
	Precision	0.801	0.807	0.818

6.5 Evaluation measures

The type of classification we carried out on Twitter is a typical form of a binary classification in which the input, Tweet, is to be classified into one, and only one, of two non-overlapping classes (positive, negative). There exist a number of performance measures used in binary classifiers in different areas of application such as F-score, precision, recall, and specificity.

Opinion or sentiment mining deals with meanings that are most of the time indirect (implied) and complex (opinions and emotions are not easy to interpret from text). So far, there is no consensus on the choice of measures used to evaluate the performance of classifiers in opinion, subjectivity, and sentiment analysis (Sokolova and Lapalme 2009). However, we found that most of the work on sentiment analysis uses accuracy as the measure of overall effectiveness of a classifier in sentiment analysis (Lee and Pang 2008; Go et al. 2009; Speriosu et al. 2011; Pang et al. 2002.) We have added two more useful metrics, precision and recall, that measure class agreement of the data labels with the positive labels given by the classifier and effectiveness of a classifier to identify positive labels, respectively. Our results are discussed in the next section.

7 Discussion and comparison with previous work

Based on the results, we observed that semantic feature vectors (SFV) consistently achieved the best accuracy with different classifiers (SVM and MNB) on the Stanford testing

dataset compared to the polarity and frequency feature vectors, using the full feature space as illustrated in Table 7, or using selected features as illustrated in Table 8. The TF/IDF feature vectors, however, outperformed the semantic feature vectors using the full feature space. Yet, the semantic feature vectors significantly outperformed the TF/IDF feature vectors on selected features.

Using the larger STS-Gold dataset, the semantic feature vectors (SFV) achieved slightly better or comparable results to the baselines as illustrated in Table 9 with the full feature space. However, by using selected features, the semantic feature vectors (SFV) significantly outperformed all the baselines using different classifiers, SVM, BNB, and MNB. It is worth noticing that with a larger dataset, the classification accuracy drops significantly with the TF/IDF-based datasets. On the other hand, with the SFV-based datasets, the classification accuracy remains consistent at acceptable levels. Consistent levels of accuracies are desirable especially in sentiment analysis of social networks since the size of data is usually very large. Moreover, it can be observed that the semantic feature vectors (SFV) always achieved the best results with significant improvement in accuracy with highly correlated set of features with the class label, i.e., the features that are expected to strongly define the semantics of the Tweet. Other dataset representations, such as polarity, do not exhibit comparable improvement.

Our results compared favorably with other research work conducted on similar datasets. Go et al. (2009) achieved the maximum accuracy of 83% using MaxEnt trained on a combination of unigrams and bigrams using the Stanford dataset. Our method outperformed the original results produced by Go et al. with maximum accuracy of 84.96% using SVM classifier. Among other research works that compared their results with the Stanford STS dataset, Speriosu et al. (2011) tested a subset of the Stanford Twitter Sentiment test set with 75 negative and 108 positive Tweets. They reported the best accuracy of 84.7% using label propagation on a rather complicated graph that has users, Tweets, word unigrams, word bigrams, hashtags, and emoticons as its nodes. Furthermore, our results outperformed Speriou's results using a simpler logic.

8 Conclusion

Twitter is one of the most popular social networks where users can express their opinions about a countless number of topics. This wealth of public opinion attracts vast interest in sentiment analysis of Twitter data. Machine learning approaches for sentiment analysis rely on feature vectors extraction to represent the most relevant and important text features (Cambria et al. 2013) that can be used to train classifiers, such as naïve Bayes (NB) and support vector machines (SVMs). Feature vector extraction eliminates many semantic relationships in the text. Yet, in many cases the sentiment conveyed by a word is implicitly associated with the semantics of its context (Cambria et al. 2013). Several methods reported in the literature for incorporating semantics in sentiment analysis suffer from a number of drawbacks including costly manual intervention, domain dependence, and limited predefined knowledge bases.

In this paper, we proposed a new fully automated, domain-independent method for constructing Twitter feature vectors for sentiment classification by using sentiment replacement and a custom fuzzy thesaurus. Sentiment replacement of common Twitter's slogans and abbreviations yielded a significant reduction in the original feature space dimensionality by nearly 35% on the STS data. The experimental results showed that the semantic feature vectors (SFV) consistently produced better results than the baselines. Furthermore, comparison with previous work showed that the proposed method outperformed other methods reported in the literature using the same benchmark data.

Our proposed future work includes expanding the scope of our experiments by exploring the impact of using other types of thesauri on more diverse datasets. In this research, we implemented a custom thesaurus based on a Twitter corpus. In the future, a potential extension is to experiment with a general thesaurus based on a common corpus. In addition, assessing the effectiveness of semantic feature vectors with different languages, such as Arabic, would provide insights on the generality of the approach. Moreover, we plan to test the effectiveness of the proposed approach in other text mining applications, such as semantic recommender systems.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Research involving human participants and/or animals Not applicable.

Informed consent Not applicable.

References

Abbasi A, Chen H, Salem A (2008) Sentiment analysis in multiple languages: features selection for opinion classification in web forums. ACM Trans Inf Syst (TOIS) 26(3):1–34

Agarwal A, Xie B, Vovsha I, Rambow O (2011) Sentiment analysis of Twitter data. In: Proceedings of the workshop on languages in social media. Association for Computational Linguistics, pp 30–38

Barbosa L, Feng J (2010) Robust sentiment detection on Twitter from biased and noisy data. In: 23rd International conference on computational linguistics. Association for Computational Linguistics, pp 36–44

Batra S, Rao D (2010) Entity based sentiment analysis on Twitter. Science 9(4):1–12

Bhuta S, Doshi A, Doshi U, Narvekar M (2014) A review of techniques for sentiment analysis of Twitter data. In: International conference on issues and challenges in intelligent computing techniques (ICICT). IEEE, pp 583–591

Boulianne S (2015) Social media use and participation: a meta-analysis of current research. Inf Commun Soc 18(5):524–538

Cambria E, Schuller B, Xia Y, Havasi C (2013) New avenues in opinion mining and sentiment analysis. IEEE Intell Syst 28:15–21

Cambria E, Speer R, Havasi C, Hussain A (2010) SenticNet: a publicly available semantic resource for opinion mining. AAAI fall symposium: commonsense knowledge 10

Elfeky M, Elhawary M (2010) Mining Arabic business reviews. In: International conference in data mining. IEEE, Sydney. pp 1108–1113

Esuli A (2006) SentiWordNet: a publicly available lexical resource for opinion mining. In: Proceedings of the 5th conference on language resources and evaluation, pp 417–422 (2006)

Garcia I, Ng YK (2006) Eliminating redundant and less-informative RSS news articles based on word similarity and a fuzzy equivalence relation. In: Tools with artificial intelligence, ICTAI'06. IEEE, pp 465–473

Go A, Bhayani R, Huang L (2009). Twitter sentiment classification using distant supervision. Stanford digital library technologies projects

Hotho A, Nürnberger A, Paaß G (2005) A brief survey of text mining. Ldv Forum 20(1):19–62

Ismail HM (2014) Using concept maps and fuzzy set information retrieval model to dynamically personalize RSS feeds. Int J Comput Sci Netw Secur 14(2):10

Ismail HM, Harous S, Belkhouche B (2016) A comparative analysis of machine learning classifiers for Twitter sentiment analysis. Res Comput Sci 110:71–83

Ismail HM, Zaki N, Belkhouche B (2016) Using custom fuzzy thesaurus to incorporate semantics and reduce data sparsity for Twitter sentiment analysis. In: 3rd International conference on soft computing and machine intelligence (ISCMI). IEEE, pp 47–52

Jiang L, Yu M, Zhou M, Liu X, Zhao T (2011) Target-dependent Twitter sentiment classification. In: Annual meeting of the association for computational linguistics. Association for Computational Linguistics, Portland, pp 151–160

Kao A, Poteet SR (eds) (2007) Natural language processing and text mining. Springer, Berlin

Kontopoulos E, Berberidis C, Dergiades T, Bassiliades N (2013) Ontology-based sentiment analysis of Twitter posts. Expert Syst Appl 40(10):4065–4074

Kraft DH, Bordogna G, Pasi G (1999) Fuzzy set techniques in information retrieval. Fuzzy Sets Approx Reason Inf Syst 5(6):469–510

Lee B, Pang L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 2(1–2):1–135

Lima ACE, de Castro LN, Corchado JM (2015) A polarity analysis framework for Twitter messages. Applied Mathematics and Computation 270(1):756–767

Liu Y, Kliman-Silver C, Mislove A (2014) The Tweets They Are a-Changin: Evolution of Twitter Users and Behavior. ICWSM 30:5–314

LOL, OMG and ILY: 60 of The Dominating Abbreviations (2014) (Just English) Retrieved November 2015, from http://justenglish.me/2014/07/18/lol-omg-and-ily-60-of-the-dominating-abbreviations/

Manning CD, Raghavan P, Schütze H (2009) Text classification and naive bayes. In: Introduction to information retrieval. Cambridge University Press, pp 253–287

Ogawa Y, Morita T, Kobayashi K (1991) A fuzzy document retrieval system using the keyword connection matrix and a learning method. Fuzzy Sets Syst 39(2):163–179

Pang B, Lee L, Vaithyanathan S (2002) Thumbs up: sentiment classification using machine learning techniques. Association for Computational Linguistics, Stroudsburg

Perez-Tellez F, Pinto D, Cardiff J, Rosso P (2010) On the difficulty of clustering company Tweets. In: 2nd International workshop on search and mining user-generated contents. ACM, New York, pp 95–102

Pew Research Center. (2014, November). Cell Phones, Social Media, and Campaign 2014. (Pew Research Center) Retrieved January 2016, from http://www.pewinternet.org/2014/11/03/cell-phones-social-media-and-campaign-2014

Porter MF (1980) An Algorithm for Suffix Stripping. Program 14(3):130–137

Saif H, Fernandez M, He Y, Alani H (2013) Evaluation datasets for Twitter sentiment analysis a survey and a new dataset, the STS-gold. In: Interantional workshop on emotion and sentiment in social and expressive media: approaches and perspectives from AI (ESSEM 2013). Italy

Saif H, He Y, Alani H (2012) Alleviating data sparsity for twitter sentiment analysis. Making sense of microposts. CEUR-WS. org, Lyon, France

Saif H, He Y, Fernandez M, Alani H (2016) Contextual semantics for sentiment analysis of twitter. Inf Process Manag 52(1):5–19

Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45(4):427–437

Speriosu M, Sudan N, Upadhyay S, Baldridge J (2011) Twitter polarity classification with label propagation over lexical links and the follower graph. In: Conference on empirical methods in natural language processing. UK, pp 53–63

Strapparava C, Valitutti A (2004) WordNet affect: an affective extension of WordNet. LREC 4:1083–1086

Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. Comput Linguist 37:267–307

Turney PD, Littman ML (2003) Measuring praise and criticism: inference of semantic orientation from association. ACM Trans Inf Syst 21(4):315–346

Vapnik VN, Vapnik V (1998) Statistical learning theory. Wiley, New York

Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: International conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, Vancouver, pp 347–354

Witten IH, Frank E, Hall MA, Pal CJ (2016) Data mining: practical machine learning tools and techniques. Morgan Kaufmann, Burlington

Yerra R, Ng YK (2005) Detecting similar HTML documents using a fuzzy set information retrieval approach. In: Granular computing IEEE International Conference, IEEE. 2:693–699

Zadeh LA (1965) Fuzzy Sets. Inf Control 8:338–353

Zaki N, Lazarova-Molnar S, El-Hajj W, Campbell P (2009) Protein-protein interaction based on pairwise similarity. BMC Bioinf 10(1):150

Zhou P, Chaovalit L (2005) Movie review mining: a comparison between supervised and unsupervised classification approaches. In: International conference on system sciences. IEEE, Hawaii, pp 112c–112c