FEATURE SET ENSEMBLES FOR SENTIMENT ANALYSIS OF TWEETS

PREPRINT

D. Griol, C. Kanagal-Balakrishna, Z. Callejas
Universidad Carlos III de Madrid
Universidad de Granada

This is a pre-print version of the chapter: Griol, D., Kanagal-Balakrishna, C., Callejas, Z. (2021). Feature Set Ensembles for Sentiment Analysis of Tweets. In: Phillips-Wren, G., Esposito, A., Jain, L.C. (eds) Advances in Data Science: Methodologies and Applications. Intelligent Systems Reference Library, vol 189. Springer, Cham. https://doi.org/10.1007/978-3-030-51870-7_10 (https://link.springer.com/chapter/10.1007/978-3-030-51870-7_10)

ABSTRACT

In recent years, sentiment analysis has attracted a lot of research attention due to the explosive growth of online social media usage and the abundant user data they generate. Twitter is one of the most popular online social networks and a microblogging platform where users share their thoughts and opinions on various topics. Twitter enforces a character limit on tweets, which makes users find creative ways to express themselves using acronyms, abbreviations, emoticons, etc. Additionally, communication on Twitter does not always follow standard grammar or spelling rules. These peculiarities can be used as features for performing sentiment classification of tweets. In this chapter, we propose a Maximum Entropy classifier that uses an ensemble of feature sets that encompass opinion lexicons, n-grams and word clusters to boost the performance of the sentiment classifier. We also demonstrate that using several opinion lexicons as feature sets provides a better performance than using just one, at the same time as adding word cluster information enriches the feature space.

1 Introduction

Due to the explosive growth of online social media in the last few years, people are increasingly turning to social media platforms such as Facebook, Twitter, Instagram, Tumblr, LinkedIn, etc., to share their thoughts, views and opinions on products, services, politics, celebrities, events, and companies. This has resulted in a massive amount of user-generated data [1].

As the usage of online social media has grown, so has the interest in the field of sentiment analysis [2, 3, 4]. For the scientific community, sentiment analysis is a challenging and complex field of study with applications in multiple disciplines and has become one of the most active research areas in Natural Language Processing, data mining, web mining and management sciences. For industry, the massive amount of user-generated data is fertile ground for extracting consumer opinion and sentiment towards their brands. In recent years, we have seen how social media has helped reshape businesses and sway public opinion and sentiment, sometimes with a single viral post or tweet. Therefore, monitoring public sentiment towards their products and services enables them to cater to their customers better.

In the last few years, Twitter has become a hugely popular microblogging platform with over 500 million tweets a day. However, Twitter only allows short messages of up to 140 characters which results in users using abbreviations, acronyms, emoticons, etc., to better express themselves. The field of sentiment analysis in Twitter therefore includes the various complexities brought by this form of communication using short informal text. The main motivation for studying sentiment analysis in Twitter is due to the immense academic as well as commercial value that it provides [5, 6, 7].

Besides its commercial applications, the number of application-oriented research papers published on sentiment analysis has been steadily increasing. For example, several researchers have used sentiment information to predict movie success and box-office revenue. Mishne and Glance showed that positive sentiment is a better predictor of movie success than simple buzz count [8]. Researchers have also analyzed sentiments of public opinions in the context of electoral politics. For example, in [9], a sentiment score was computed based simply on counting positive and negative sentiment words, which was shown to correlate well with presidential approval, political election polls, and consumer confidence surveys. Market prediction is also another popular research area for sentiment analysis [10].

The main research question that we want to ask in this chapter is: *Can we combine different feature extraction methods to boost the performance of sentiment classification of tweets?*

Raw data cannot be fed directly to the algorithms themselves as most of the algorithms expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. Feature extraction is the process of transforming text documents into numerical feature vectors. There are many standard feature extraction methods for sentiment analysis of text data such as Bag of Words representation, tokenization, etc. Since feature extraction usually results in high dimensionality of features, it is important to use features that provide useful information to the machine learning algorithm.

Sub-question 1: Does extracting features using opinion lexicons add value to the feature space?

Opinion Lexicons refers to a list of opinion words such as good, excellent, poor, bad, etc., which are used to indicate positive and negative sentiment. The positive and negative sentiment scores of each tweet can be extracted as features using Opinion Lexicons. We investigate if Opinion Lexicons boost the performance of sentiment classification of tweets.

Sub-question 2: Does using word clusters as features add value to the feature space

Word clustering is a technique for partitioning sets of words into subsets of semantically similar words, for example, Monday, Tuesday, Wednesday, etc., would be included in a word cluster together. Word clusters can be used as features themselves. Thus, word clustering has a potential to reduce sparsity of the feature space. We investigate if using word clusters as features improves the performance of sentiment classification of tweets.

The remainder of the chapter is as follows. Section 2 describes the motivation of our proposal and related work. Section 3 summarizes the basic terminology, levels and approaches for sentiment analysis. Section 4 describes the main data sources used in our research. Section 5 presents the experimental process that we have followed, the feature sets and results of the evaluation. Finally, Section 6 presents the conclusions and suggests some future work guidelines.

2 State of the art

Sentiment Analysis can be defined as a field of study consisting of a series of methods, techniques, and tools about detecting and extracting subjective information of people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes expressed in written text [10, 11]. Though there are some nuances in the definition of the terms as well as their applications, for our study, we will treat sentiment analysis, opinion mining, subjectivity analysis, opinion analysis, review mining, opinion extraction, etc., interchangeably.

Traditionally, the desired practical outcome of performing sentiment analysis on text is to classify the polarity of the opinion. Opinion polarity can be classified into 3 categories, i.e., if the opinion expressed in the text is positive, negative or neutral towards the entity.

An important part of our information-gathering behavior has always been to find out what other people think. With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others [12]. It is not always feasible for potential customers to go to a physical store to examine the features and performance of various products. It is also difficult to predict how the products will hold up over time. The general trend now before selecting a product and making a purchase is to read the reviews, blog posts, etc., written by other customers about their experiences with the product to better gauge if it will be a good fit in accordance to their product requirements.

Factors that further advanced sentiment analysis during the last decade are:

- The rise of machine learning methods in natural language processing and information retrieval;
- The availability of datasets for machine learning algorithms to be trained on, due to the World Wide Web and, specifically, the development of review-aggregation web-sites;
- Realization of the intellectual challenges and commercial and intelligence applications that the area offers [12].
- Evolution of the web from Web 1.0 to Web 2.0. Web 2.0 is an evolution from passive viewing of information to interactive creation of user generated data by the collaboration of users on the Web. The evolution of Web from Web 1.0 to Web 2.0 was enabled by the rise of read/write platforms such as blogging, social networks, and free image and video sharing sites. These platforms have jointly allowed exceptionally effortless content creation and sharing by anyone [13].

With the proliferation of Web 2.0 applications, research field of sentiment analysis has been progressing rapidly due to the vast amounts of data generated by such applications. Blogs, review sites, forums, microblogging sites, wikis and social networks have all provided different dimensions to the data used for sentiment analysis.

3 Basic Terminology, levels and approaches of Sentiment Analysis

Formally, Sentiment Analysis is the computational study of opinions, sentiments and emotions expressed in text. The goal of sentiment analysis is to detect subjective information contained in various sources and determine the mind-set of an author towards an issue or the overall disposition of a document. The analysis is done on user generated content on the Web which contains opinions, sentiments or views. An opinionated document can be a product review, a forum post, a blog or a

tweet, that evaluates an object. The opinions indicated can be about anything or anybody, for e .g. products, issues, people, organizations or a service [13].

Mathematically, Liu defines an opinion as a quintuple, (e, a, s, h, t), where e is the target entity; also known as object, a is the target aspect of entity e on which the opinion has been given; also known as feature of the object, s is the sentiment of the opinion on aspect a of entity e, h is the opinion holder, and t is the opinion posting time [10] .

- Object: An entity which can be a product, person, event, organization, or topic. The object can have attributes, features or components associated with it. Further on the components can have subcomponents and attributes
- Feature: An attribute (or a part) of the object with respect to which evaluation is made.
- Opinion orientation or polarity: The orientation of an opinion on a feature indicates whether the opinion is positive, negative or neutral. It can also be a rating (e.g., 1-5 stars). Most work has been done on binary classification i.e. into positive or negative. But opinions can vary in intensity from very strong to weak. For example a positive sentiment can range from content to happy to ecstatic. Thus, strength of opinion can be scaled and depending on the application the number of levels can be decided.
- Opinion holder: The holder of an opinion is the person or organization that expresses the opinion [13].

Sentiment Analysis can be performed at different structural levels, ranging from individual words to entire documents. Depending on the granularity required, Sentiment Analysis Research has been mainly carried out at three levels namely: Document Level, Sentence Level and Aspect Level.

Document level Sentiment Analysis is the simplest form of classification. The whole document is considered as a basic unit of information. The task at the document level is to classify whether the whole document expresses a positive, negative or neutral sentiment. However, there are two assumptions to be made. Firstly, this level of analysis assumes that the entire document expresses opinions on a single entity (film, book, hotel, etc.). Secondly, it is assumed that the opinions are from a single opinion holder. Thus, document level Sentiment Analysis is not applicable to documents that evaluate or compare opinions on multiple entities [10].

Sentence level Sentiment Analysis aims to go to the sentences and determine whether each sentence expresses a positive, negative or neutral opinion. Neutral usually means no opinion. Sentence level classification assumes that the sentence expresses only one opinion, which is not true in many cases. Sentence level classification is closely related to subjectivity classification which distinguishes sentences which provide factual information from sentences that express subjective opinions. The former is called an objective sentence, while the latter is called a subjective sentence [10, 14, 15]. Therefore, the first task at this level is to determine if the sentence is opinionated or not, i.e., subjective or objective. The second task is to determine the polarity of the sentence, i.e., positive, negative or neutral.

Aspect level sentiment analysis is based on the idea that an opinion consists of a sentiment, i.e., positive, negative or neutral, as well as a target of the opinion, aspect. Aspect level sentiment analysis performs a finer-grained analysis compared to document level and sentence level sentiment analysis. The goal of this level of analysis is to discover sentiments on entities and/or their aspects. Thus, aspect level sentiment analysis is a better representation when it comes to texts such as product reviews which usually involve opinions on multiple aspects.

There are two well-established approaches to carrying out sentiment analysis. One is the lexicon-based approach where the classification process relies on the rules and heuristics obtained from linguistic knowledge. The other is the machine-learning approach where algorithms learn underlying information from previously annotated data which allows them to classify new unlabeled data. There have also been a growing number of studies which have successfully implemented a hybrid approach by combining lexicon-based approach and machine-learning approach.

The lexicon-based approach depends on finding the opinion lexicon which can be used to analyze the text. There are two methods in this approach; dictionary-based approach and corpus based approach. The dictionary based approach depends on finding opinion seed words, and then searching the dictionary for their synonyms and antonyms. On the other hand, the corpus based approach begins with a seed list of opinion words, and then finds other opinion words in a large corpus to help in finding opinion words with context specific orientations. This can be accomplished using statistical or semantic methods [16].

In the dictionary-based approach, a small set of opinion words is collected manually with known prior polarity or sentiment orientations. Then, this seed set is expanded by searching in a well-known corpora such as WordNet or a thesaurus for their synonyms and antonyms. The newly found words are added to the seed list then the next iteration starts. The iterative process stops when no new words are found. After the process is completed, manual inspection can be carried out to remove or correct errors [16, 11].

Corpus based methods rely on syntactic or statistical techniques like co-occurrence of word with another word whose polarity is known. For this approach, [17] used a corpus and some seed adjective sentiment words to find additional sentiment adjectives in the corpus. Their technique exploited a set of linguistic rules or conventions on connectives to identify more adjective sentiment words and their orientations from the corpus.

Using the corpus-based approach alone is not as effective as the dictionary-based approach because it is hard to prepare a huge corpus which covers all English words. However, the advantage of corpus-based approach is that it can help to find domain and context specific opinion words and their orientations using a domain corpus [16]. But it is important to note that having a sentiment lexicon (even with domain specific orientations), does not mean that a word in the lexicon always expresses an opinion/sentiment in a specific sentence. For example, in "I am looking for a good car to buy", "good" here does not express either a positive or negative opinion on any particular car. Due to contributions of many researchers, several general-purpose subjectivity, sentiment, and emotion lexicons have been constructed and are also publicly available [18, 16].

The text classification methods using Machine Learning approach can be roughly divided into supervised and unsupervised learning methods. The supervised methods make use of a large number of labeled training documents. The unsupervised methods are used when it is difficult to find these labeled training documents. Machine learning approach relies on Machine Learning algorithms to solve the problem of sentiment classification. To achieve this, the machine learning approach treats sentiment analysis as a regular text classification problem, where instead of classifying documents of different topics (e.g., politics, sciences, and sports), we estimate positive, negative, and neutral classes [19].

The goal of the supervised machine learning approach is to predict and classify the sentiment of a given text based on information learned from past examples. The supervised learning methods, therefore, depend on the existence of labeled training documents. To build the classification model, training data with annotated sentiment is applied to the chosen supervised machine learning classifier. Then, the unlabeled testing data which is not used for training is applied to the trained classifier model. With the results obtained, sentiment polarity of the test data is predicted. Typical classifiers used in this approach include: probabilistic classifiers, linear classifiers, decision trees classifiers, and rule-based classifiers.

Probabilistic classifiers are among the most popular classifiers used in the machine learning community and increasingly in many applications. These classifiers are derived from generative probability models which provide a principled way to the study of statistical classification in complex domains such as natural language and visual processing. Probabilistic classification is the study of approximating a joint distribution with a product distribution. Bayes rule is used to estimate the conditional probability of a class label, and then assumptions are made on the model, to decompose this probability into a product of conditional probabilities [20]. Three of the most famous probabilistic classifiers are Naive Bayes classifiers, Bayesian Network and Maximum Entropy classifiers.

There are many kinds of linear classifiers, among which Support Vector Machines is popularly used for text data. These classifiers are supervised machine learning models used for binary classification and regression analysis. However, research studies have proposed various approaches to handle multiclass classification using SVM. Support vector machines (SVMs) are highly effective for traditional text categorization, and can outperform Naive Bayes [12].

Decision trees are based on a hierarchical decomposition of the training data, in which a condition on the attribute value is used in order to divide the data space hierarchically. The division of the data space is performed recursively in the decision tree, until the leaf nodes contain a certain minimum number of records, or some conditions on class purity. The majority class label in the leaf node is used for the purposes of classification. For a given test instance, the sequence of predicates is applied at the nodes, in order to traverse a path of the tree in top-down fashion and determine the relevant leaf node.

In rule-based classifiers, the data space is modeled with a set of rules, in which the left hand side is a condition on the underlying feature set, and the right hand side is the class label. The rule set

is essentially the model which is generated from the training data. For a given test instance, we determine the set of rules for which the test instance satisfies the condition on the left hand side of the rule. We determine the predicted class label as a function of the class labels of the rules which are satisfied by the test instance. Rule-based Classifiers are related to the decision tree classifiers because both encode rules on the feature space. The main difference is that the decision tree classifier uses the hierarchical approach, whereas the rule-based classifier allows for overlap in the decision space [21]. In these classifiers, the training phase generates the rules based on different criteria. Two of the most common conditions which are used for rule generation are those of support and confidence.

Classifier ensemble have been also proposed to combine different classifiers in conjunction with a voting mechanism in order to perform the classification. The basis is that since different classifiers are susceptible to different kinds of overtraining and errors, a combination classifier is likely to yield much more robust results. This technique is also sometimes referred to as stacking or classifier committee construction. Ensemble learning has been used quite frequently in text categorization. Most methods simply use weighted combinations of classifier outputs (either in terms of scores or ranks) in order to provide the final classification result. The major challenge in ensemble learning is to provide the appropriate combination of classifiers for a particular scenario. This combination can significantly vary with different scenarios and data sets [6, 21].

4 Data sources

The dataset chosen to build a classifier for sentiment analysis can have a significant impact on the performance of the classifier when implemented on the test data. Several important factors need to be considered before choosing a dataset. When it comes to analyzing tweets, we need to consider the effect of the domain focused tweets, data structure as well as the objective of the classification.

Twitter Sentiment Analysis SemEval Task B Dataset was chosen for experimentation using various classification methods. To remedy the lack of datasets which is hindering sentiment analysis research, Nakov et al. [22] released a twitter training dataset to the research community to be used for evaluation and comparison between approaches. The SemEval Tweet corpus contains tweets with sentiment with sentiment expressions annotated with overall message-level polarity. The tweets that were gathered express sentiment about popular topics. The collection of tweets span over a one-year period from January 2012 to January 2013. Public streaming Twitter API was used to download tweets.

The dataset was annotated for sentiment on Mechanical Turk, a crowdsourcing marketplace that enables individuals or businesses to use human intelligence to perform tasks that computers are currently unable to do such as image recognition, audio transcription, machine learning algorithm training, sentiment analysis, data normalization, surveys, etc., in exchange for a reward[1]. Each sentence was annotated by five Mechanical Turk workers. They had to indicate the overall polarity of the sentence as positive, negative or neutral as well as the polarity of a subjective word or phrase. However, the dataset used to build our classifier only contains annotations of overall message-level polarity. The final polarity of the entire sentence was determined based on the majority of the labels [22].

SemEval Twitter Corpus consists of 13,541 tweets (or instances) collected between January 2012 and January 2013. The domain of the tweets is not indicated in [22]. Each instance in the corpus contains values for two attributes namely; Content and Class. The instances of the content attribute contain the tweets themselves containing data in a string format. The instances of the class attribute contain three nominal values (classes) namely positive, negative and neutral. It should be noted that, the turkers were instructed to choose the stronger sentiment in messages conveying both positive and negative sentiments. Table 1 illustrates the distribution of tweets from the corpus as well as an example tweet and its class as labeled by the turkers.

We see from Figure 1 that the class distribution is not balanced. For model training and classification, balanced class distribution is very important to ensure the prior probabilities are not biased caused by the imbalanced class distribution.

There are many methods to address the class imbalance problem such as collecting more data, changing the performance metric, resampling the dataset, generating synthetic samples, penalized models, etc. In order to balance the dataset, we are going to implement resampling of the dataset. Since the class with the lowest number of instances (Negative) still has a considerable number of instances which can be used to train the classifiers, we are going to perform random sampling without

[1]Amazon Mechanical Turk, https://www.mturk.com/

Table 1: Examples from SemEval Twitter Corpus

Class	Count	Example
Positive	5,232	Gas by my house hit $3.39!!!! I am going to Chapel Hill on Sat :)
Negative	6,242	Theo Walcott is still shit, watch Rafa and Johnny deal with him on Saturday.
Neutral	2,067	Fact of the day; Halloween night is Papa John's second busiest night of the year behind Super Bowl Sunday.

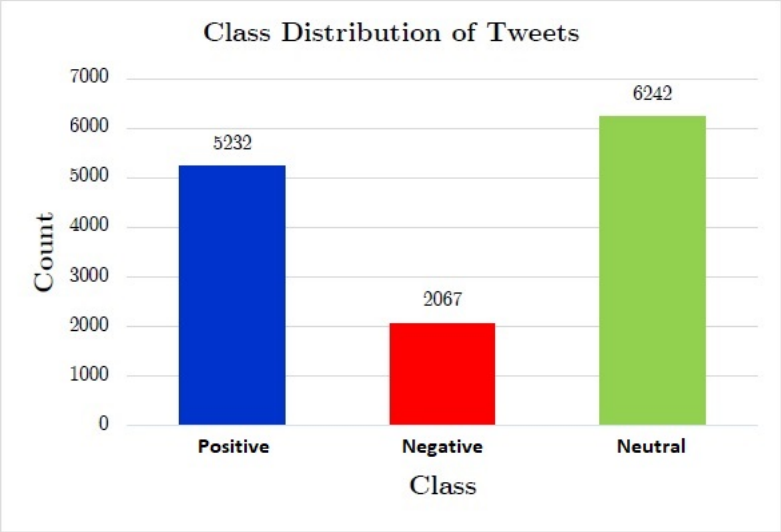

Figure 1: Class Distribution of tweets

replacement so that instances from the over-represented classes are removed from the dataset. Figure 2 illustrates the class distribution by tweets after sampling without replacement.

4.1 Sentiment Lexicons

Sentiment Lexicons, also known as Opinion Lexicons, refers to a list of opinion words such as good, excellent, poor, bad, etc., which are used to indicate positive and negative sentiment. Opinion Lexicons play an important role in extracting two very important features; positive and negative sentiment scores. Extraction of these features could enhance the accuracy of the classification system and the frequency of these sentiment words directly maps to overall sentiment of a tweet. Therefore, we can enrich the feature space with opinion lexicon information, where each tweet (or instance) as the associated positive and negative sentiment score.

The AFINN lexicon is based on the Affective Norms for English Words lexicon (ANEW) proposed in [23]. ANEW provides emotional ratings for a large number of English words. These ratings are calculated according to the psychological reaction of a person to a specific word, being the valence the most useful value for sentiment analysis. Valence ranges in the scale pleasant-unpleasant. This lexicon was released before the rise of microblogging and therefore does not contain the common slang words used on microblogging platforms such as Twitter. Nielsen created the AFINN lexicon [24], which is more focused on the language used in microblogging platforms. The word list includes slang and obscene words as well as acronyms and web jargon. Positive words are scored from 1 to 5 and negative words from -1 to -5, reason why this lexicon is useful for strength estimation. The lexicon includes 2,477 English words [25].

The AFINN lexicon extracts two features from each tweet (or instance). AFINN Positivity Score and AFINN Negativity Score, that are the sum of the ratings of positive and negative words of the tweet that matches the AFINN lexicon, respectively.

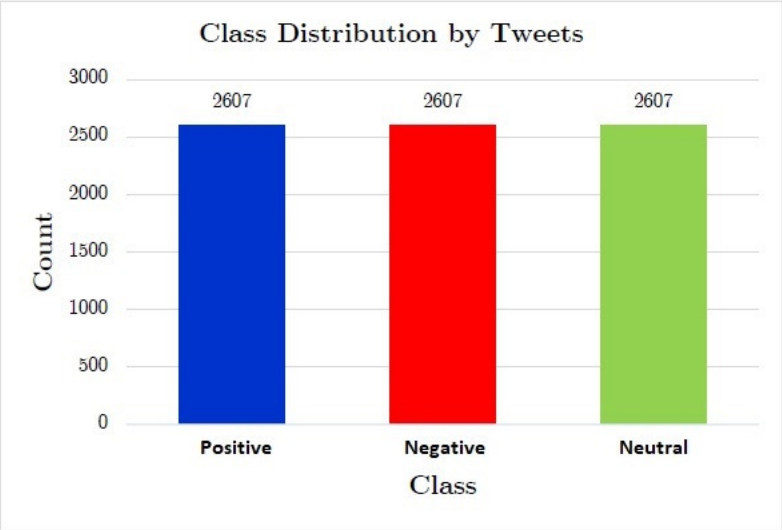

Figure 2: Class Distribution of tweets after sampling without replacement

The Bing Liu Opinion lexicon is one of the most widely used sentiment lexicons for sentiment analysis. Hu and Liu [26] proposed a lexicon-based algorithm for aspect level sentiment classification, but the method can determine the sentiment orientation of a sentence as well. It was based on a sentiment lexicon generated using a bootstrapping strategy with some given positive and negative sentiment word seeds and the synonyms and antonyms relations in WordNet. The sentiment orientation of a sentence was determined by summing up the orientation scores of all sentiment words in the sentence. A positive word was given the sentiment score of +1 and a negative word was given the sentiment score of -1. Negation words and contrary words (e.g., but and however) were also considered [10]. The Lexicon includes 6,800 English words.

The Bing Liu Opinion lexicon extracts two features from the tweets (or instances). Bing Liu Positivity Score and Bing Liu Negativity Score, that are the sum of the orientation scores of positive and negative sentiment words in the tweet that matches the Bing Liu lexicon, respectively.

The NRC Word-Emotion Association Lexicon is a lexicon that includes a large set of human-provided words with their emotional tags. By conducting a tagging process in the crowdsourcing Amazon Mechanical Turk platform, Mohammad and Turney [27] created a word lexicon that contains more than 14,000 distinct English words annotated according to the Plutchik's wheel of emotions. The wheel is composed by four pairs of opposite emotion states: joy-trust, sadness-anger, surprise-fear, and anticipation-disgust. These words can be tagged to multiple categories. Additionally, NRC words are tagged according to polarity classes positive and negative [25]. The NRC Word-Emotion Association lexicon extracts ten features from the tweets (or instances) namely; NRC Joy, NRC Trust, NRC Sadness, NRC Anger, NRC Surprise, NRC Fear, NRC Anticipation, NRC Positive and NRC Negative.

NRC Word-Emotion Association Lexicon did not include expressions such as hashtags, slang words, misspelled words, etc., that are commonly seen on social media (i.e. twitter, facebook, etc.). The NRC-10 Expanded Lexicon was created to address this issue. The NRC-10 Expanded lexicon extracts ten features from the tweets (or instances): NRC Joy, NRC Trust, NRC Sadness, NRC Anger, NRC Surprise, NRC Fear, NRC Anticipation, NRC Positive and NRC Negative.

The NRC Hashtag Emotion Lexicon consists of an association of words with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) generated automatically from tweets with emotion-word hashtags such as #happy and #angry. It contains 16832 distinct English words. NRC Hashtag Emotion Lexicon extracts 8 features from the tweets (or instances) namely; NRC Joy, NRC Trust, NRC Sadness, NRC Anger, NRC Surprise, NRC Fear, NRC Anticipation.

The NRC Hashtag Sentiment Lexicon consists of an association of words with positive and negative sentiment generated automatically from tweets with sentiment-word hashtags such as #amazing and #terrible. It consists of 54,129 unigrams (words), 316,531 bigrams and 308,808 pairs. NRC Hashtag

Table 2: Models defined for the feature sets 1, 2 and 3

Feature set	Feature extraction method
FS1A	AFINN Lexicon
FS1B	Bing Liu Lexicon
FS1C	NRC-10 Word Emotion Association Lexicon, NRC-10 Expanded Lexicon, NRC Hashtag Emotion Lexico and NRC Hashtag Sentiment Lexicon, Negation
FS1D	AFINN Lexicon, Bing Liu Lexicon, NRC-10 Word Emotion Association Lexicon, NRC-10 Expanded Lexicon, NRC Hashtag Emotion Lexicon and NRC Hashtag Sentiment Lexicon, Negation
FS2A	Word Unigrams, Twokenize, Binary frequency, Frequency weighting
FS2B	Word Unigrams + Bigrams, Twokenize, Binary frequency, Frequency weighting
FS2C	Word Unigrams + Bigrams + Trigrams, Twokenize, Binary Frequency, Frequency weighting
FS3A	Cluster Unigrams, Twokenize, Binary frequency, Frequency weighting
FS3B	Cluster Unigrams-Bigrams, Twokenize, Binary frequency, Frequency weighting
FS3C	Cluster Unigrams-Bigrams-Trigrams, Twokenize, Binary frequency, Frequency weighting

Sentiment Lexicon extracts two features from the tweets (or instances) namely; NRC Positive and NRC Negative[2].

5 Experimental procedure

In this section, we describe the experimentation performed on the SemEval Twitter Corpus. As previously described, we build our baseline classifiers using the sub-feature sets from the three feature sets defined. The preceding steps such as preprocessing and feature extraction are performed on the classifiers. Feature selection will be performed only on feature set 2 and feature set 3. The proposed classifier will be trained using the feature set PFS where we combine various models from feature set 1, feature set 2 and feature set 3. All the models will be trained using the classification algorithms; Maximum Entropy and Support Vector Machines. The model is evaluated as described in section 6.6. Finally, we compare the performance metrics of our baseline classifiers with that of the proposed classifier(s).

5.1 Feature sets

We have defined three feature sets that will be tested for our baseline classifier models. These feature sets are further sub-divided into classifier models that use specific feature extraction and feature selection methods. All the models will be trained using two classification algorithms; Maximum Entropy and Support Vector machines.

In feature set 1, we make use of six Sentiment Lexicons; AFINN, Bing Liu Lexicon, NRC-10 Word Emotion Association Lexicon, NRC-10 Expanded Lexicon, NRC Hashtag Emotion Lexicon, NRC Hashtag Sentiment Lexicon and Negation to extract their respective features. The Lexicons are employed in various combinations. For data preprocessing, we reduce length of elongated words, convert to lower case and replace user mentions and URLs with generic tokens.

In feature set 2, we use a combination of word N-grams such as Unigrams, Unigrams and Bigrams, Unigrams, Bigrams and Trigrams, for feature extraction. In feature set 3, we use a combination of cluster N-grams such as Unigrams, Unigrams and Bigrams, Unigrams, Bigrams and Trigrams, for feature extraction. Additionally, we also use the Twokenize tokenizer from CMU Tweet NLP tool, binary frequency of terms as well as weighted frequency as feature extraction methods in all the models. For data preprocessing, we use negation handling, reduce length of elongated words and convert words to lower case. Table 2 shows the feature extraction method and types of features used for the different models defined for each set.

[2]NRC Emotion and Sentiment Lexicons, `http://saifmohammad.com/WebPages/AccessResource.htm`

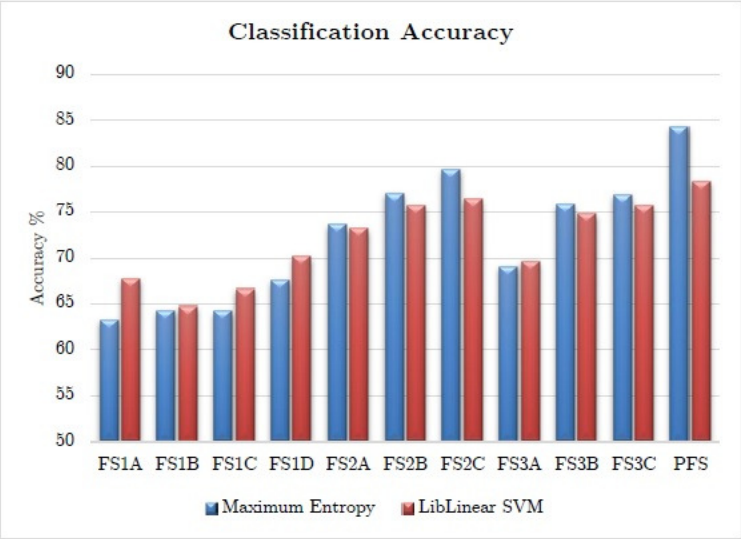

Figure 3: Classification accuracy obtained for the set of models

5.2 Results of the evaluation

Performance of classifiers is commonly measured with reference to a baseline classifier. Some of the most used baseline classifiers for text classification and sentiment analysis include Support Vector Machines, Maximum Entropy, Naive Bayes, Decision Trees and Random Forest. For the purpose of performance comparison, we consider the classifiers built using the feature set 2 model, FS2A as our primary baseline classifiers. The FS2A classifiers are built using standard data preprocessing steps such as lowering case, reducing the length of elongated words, etc. It also uses Unigrams for feature extraction which is standard for classification of tweets. The classification accuracy of models built using the feature set 1, feature set 2, feature set 3, as well as that of the proposed feature set is illustrated in Figure 3.

The classification accuracy of the baseline classifier, FS2A, which uses Maximum Entropy algorithm is 73.63%, whereas the LibLinear SVM algorithm provides an accuracy of 73.13%. While Maximum Entropy performs slightly better, the difference is not significant. When we compare the baseline classifiers which models from feature set 1, we see that none of the classifiers perform as well as the baseline classifiers for both the algorithms. Feature set 1, which uses various combinations of opinion lexicons, provides the highest classification accuracy when we combine the opinion lexicons; AFINN, Bing Liu Lexicon, NRC-10 Word Emotion Association Lexicon, NRC-10 Expanded Lexicon, NRC Hashtag Emotion Lexicon and NRC Hashtag Sentiment Lexicon with accuracies of 67.58% and 70.15% for Maximum Entropy and LibLinear SVM respectively. LibLinear SVM consistently outperforms Maximum Entropy in feature set 1.

Feature set 2 includes models built using various word n-gram combinations. FS2C Maximum Entropy classifier achieves the highest overall accuracy with 79.64%. We observe that the classification accuracy rises when we include Bigrams, Bigrams and Trigrams to the baseline classifier which only uses Unigrams. While this is true of both Maximum Entropy and LibLinear SVM, the performance improvement is more apparent with Maximum Entropy which shows a significant improvement over the baseline when the n-gram combination of unigrams, bigrams and trigrams. While LibLinear SVM shows an improvement over the unigram model, the difference between the unigram-bigram and unigram-bigram-trigram model is not significant.

Feature set 3 includes models built using various cluster n-gram combinations. FS3C Maximum Entropy classifier achieves the highest overall accuracy with 76.87%. We observe that the classification accuracy rises when we include Bigrams, Bigrams and Trigrams to the baseline classifier which only uses Unigrams. This is the case for both Maximum Entropy and LibLinear SVM, although the performance improvement is more apparent with Maximum Entropy which shows a significant improvement over the baseline when the n-gram combination of unigrams, bigrams and trigrams.

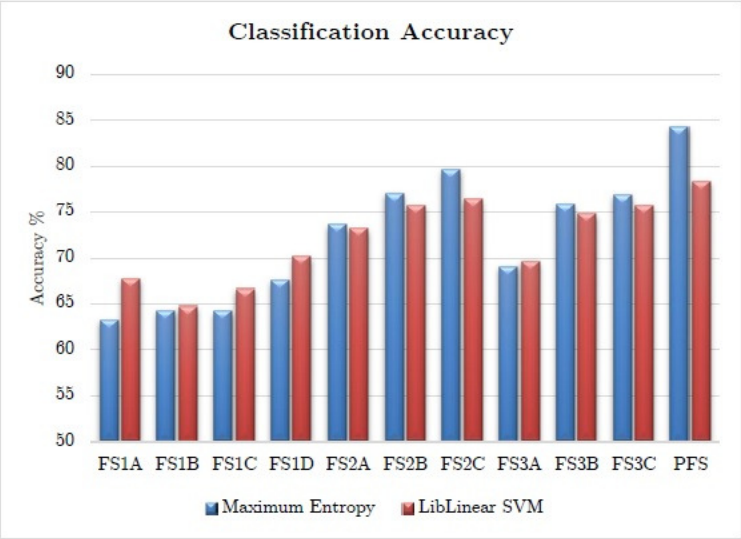

Figure 4: Kappa Statistic values obtained for the set of models

While both the algorithms show an improvement over the unigram model, the difference between the unigram-bigram and unigram-bigram-trigram model is not large.

The proposed feature set uses the best performing model among the 3 feature sets. Therefore, we combine the models FS1D, FS2C and FS3C to generate the proposed classifier model. The LibLinear SVM model achieves an accuracy of 78.32% which is better than the performance of all the other LibLinear SVM classifiers built using the 3 feature sets. However, Maximum Entropy shows a significant improvement in performance. It achieves the highest classification accuracy of 84.3% as well as the highest overall classification accuracy of all the models used. The Kappa statistic of models built using the feature sets 1, 2, and 3, as well as that of the proposed feature set is illustrated in Figure 4.

By following the guidelines of Landis and Koch [28] to interpret the Kappa statistic measures, we observe that the baseline model, FS2A, are in the 0.41 and 0.60 range which indicates a moderate strength of agreement. With feature set 1, LibLinear SVM performs better than Maximum entropy in all cases except FS1B, where Maximum Entropy and Liblinear SVM perform at the same level. FS1D performs better among all the models in feature set 1 and performs moderately well, being in the 0.41 - 0.6 range.

With feature set 2, we observe that the kappa statistic improves consistently when higher order word n-gram combinations are used for both Maximum Entropy and LibLinear SVM, with Maximum Entropy achieving the highest overall kappa measure of 0.6947 which falls in the 0.61 - 0.80 range. We can thus infer that the strength of agreement is substantial.

With feature set 3, we observe that the Kappa statistic increases with higher order cluster n-grams. Maximum Entropy outperforms LibLinear SVM, but only slightly, with a kappa statistic of 0.6531 indicating a substantial strength of agreement. The LibLinear SVM has a Kappa statistic of 0.6355 which also indicates a substantial strength of agreement.

Overall, the highest kappa statistic measure is obtained by FS2C, which includes features extracted using word unigram=bigram-trigram combination.

Figure 5 indicates the performance metrics of precision, recall and F-score for feature sets 1, 2, 3 and the proposed feature set for Maximum Entropy classifier.

For Maximum Entropy, the precision, recall and the F-score of the baseline model, FS2A, is 0.75, 0.738 and 0.739 respectively, thus having a slightly better precision compared to recall. For feature set 1, the precision ranges from 0.67 - 0.681, recall ranges from 0.632 - 0.676 and F-score ranges from 0.614 - 0.675. Thus, none of the models perform as well as the baseline model in terms of these metrics. FS1D achieves the highest precision, recall and accuracy among the feature set 1 models. For feature set 2, FS2C performs the best in terms of accuracy precision and recall achieving values of 0.803, 0.796 and 0.798 respectively. For feature set 3, FS3C performs better than the baseline

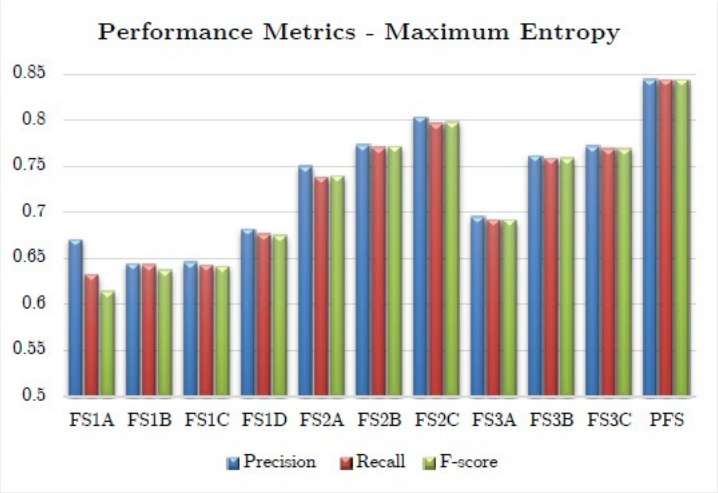

Figure 5: Performance metrics of Maximum Entropy models

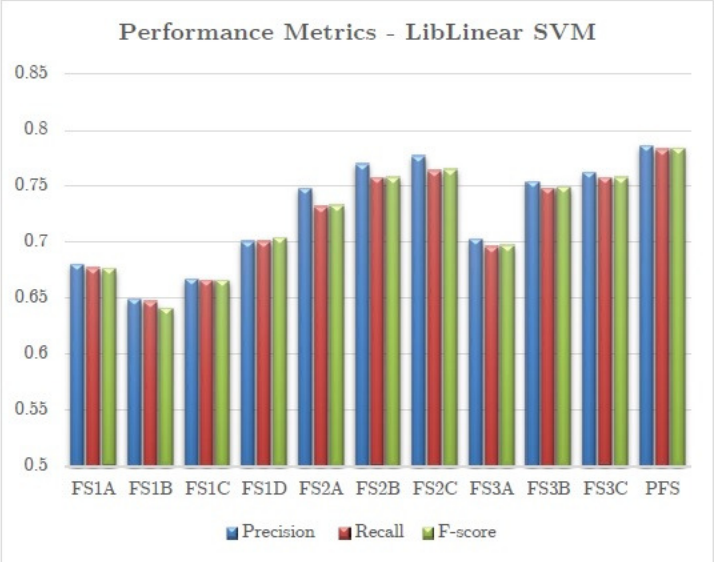

Figure 6: Performance metrics of LibLinear SVM models

values achieving 0.772, 0.769 and 0.769. PFS, model from the proposed feature set which includes cluster unigram-bigram-trigram combination, word unigram-bigram-trigram combination achieves the highest overall performance metrics compared to the baseline model with precision, recall and F-score values of 0.844, 0.843 and 0.843.

Figure 6 indicates the performance metrics of precision, recall and F-score for feature sets 1, 2, 3 and the proposed feature set for LibLinear SVM classifier.

For LibLinear SVM, the precision, recall and the F-score of the baseline model, FS2A, is 0.748, 0.732 and 0.733 respectively, thus having a slightly better precision compared to recall. For feature set 1, the precision ranges from 0.68 - 0.701, recall ranges from 0.677 - 0.701 and F-score ranges from 0.676 - 0.704. Thus, none of the models perform as well as the baseline model in terms of these metrics. FS1D achieves the highest precision, recall and accuracy among the feature set 1 models. For feature set 2, FS2C performs the best in terms of accuracy precision and recall achieving values of 0.777, 0.764 and 0.765 respectively. For feature set 3, FS3C performs the better than the baseline values achieving 0.762, 0757 and 0.758. However, we do not see a significant improvement

in the metrics for the proposed feature set model which uses LibLinear SVM compared to the other high-performing LibLinear models such as FS2C.

From our discussion, it appears that using Opinion Lexicons alone as features to train machine learning algorithms such as Maximum Entropy and Support Vector Machines does not raise classification accuracy significantly. However, using multiple Opinion Lexicons to generate features seems to provide a better performance than using them individually. Though using a standard word n-gram iteration such as unigrams to train machine learning algorithms provides a better performance than using Opinion Lexicons, adding higher order word n-grams as features significantly improves performance. However, it was observed during our experimentation that this effect only carries until trigrams.

Generating features with word n-grams of higher order than trigrams does not improve the performance and is computationally expensive since it generates a large number of features and increases sparsity. When cluster n-grams are uses ad features by themselves, they too provide a better performance with higher order n-grams. As with the word n-grams, higher order cluster n-grams provided better performance than cluster unigrams alone. And similar to word n-grams, this effect was only noticed until we reached trigrams. Using cluster n-grams of higher order not only increased the time taken for feature extraction, feature selection and model training, it also did not keep the pattern of increased performance seen with the addition of cluster bigrams and trigrams. When Opinion Lexicons, word n-grams and cluster n-grams were combined from all the high performing models of the three feature sets, Maximum Entropy classifier showed a marked improvement in performance while LibLinear SVM did not show any significant improvement.

From the different experiments, it can be concluded that a combination of word unigrams-bigrams-trigrams, cluster unigrams-bigrams-trigrams as well as a combination of six opinion lexicons used a features and then ranked using Information Gain alogorithm and the Ranker Search method provided the best performance in terms of accuracy, precision, recall, F-score and Kappa statistic when used with the Maximum Entropy Classifier with the conjugate gradient descent method.

6 Conclusions and future work

In this chapter we have presented an approach that yields improved sentiment classification of Twitter data. Sentiment classification of tweets poses a unique challenge compared to text classification performed in other mediums.

For our research, we used the SemEval Twitter Corpus which contained a large number of tweets in the neutral class compared to that of positive and negative classes. In order to reduce bias, we balanced the dataset by reducing the number of neutral tweets to that of positive and negative tweets. We explored various feature extraction methods which could enrich the feature model space such that problems of sparsity commonly associated with datasets that have a large number of attributes, such as Twitter data, is addressed.

Our major contributions are four-folds. We extensively study various feature extraction methods individually and combined using a supervised machine learning approach. First, we demonstrated that using a combination of opinion lexicons to extract features improves the sentiment classification accuracy than using an individual opinion lexicon by itself. Second, we demonstrated that using unigram-bigram-trigram Bag of Words feature improves the sentiment classification accuracy than using lower order n-gram features alone. Third, we demonstrate that when using brown word clusters as features by themselves, unigram-bigram-trigram clusters provide an improvement in performance than lower order cluster n-grams. And fourth, we proposed a classifier model which significantly raises the classification accuracy by combining various feature extraction methods. We demonstrated that by taking the external knowledge of a word cluster into account while classifying sentiment of tweets improves the performance of the classifier using a machine-based learning algorithm.

The proposed classifier uses a combination of six mainstream opinion lexicons, unigram-bigram-trigram Bag of Words and unigram-bigram-trigram clusters as features. The dimensionality of the features was reduced by feature extraction methods such as information gain algorithm and the ranker search method. Using the Multinomial Logistic Regression algorithm (Maximum Entropy) with conjugate gradient descent with the proposed set of features not only improved the accuracy over the baseline Unigram Bag of Words model by 10.67%, but still maintained a comparable training time.

As future work, additional studies need to be undertaken to determine if the results obtained can be generalized to other domains which use short informal text for communication such as Tumblr, SMS, Plurk, etc.

7 Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823907 (MENHIR: Mental health monitoring through interactive conversations `https://menhir-project.eu`).

References

[1] My T. Thai, Weili Wu, and Hui Xiong. *Big Data in Complex and Social Networks*. Chapman and Hall/CRC, 2016.

[2] L.-C. Yu, J.-L. Wu, P.-C. Chang, and H.-S. Chu. Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news. *Knowledge-Based Systems*, 41:89–97, 2013.

[3] D. Wang, S. Zhu, and T. Li. SumView: a Web-based engine for summarizing product reviews and customer opinions. *Expert Systems with Applications*, 40(1):27–33, 2013.

[4] A. Montoyo, P. Martínez-Barco, and A. Balahur. Subjectivity and sentiment analysis: an overview of the current state of the area and envisaged developments. *Decision Support Systems*, 53(4):675–679, 2012.

[5] Fazeel Abid, Muhammad Alam, Muhammad Yasir, and Chen Li. Sentiment analysis through recurrent variants latterly on convolutional neural network of twitter. *Future Generation Computer Systems*, 95:292–308, 2019.

[6] Ankit and Nabizath Saleena. An ensemble classification system for twitter sentiment analysis. *Procedia Computer Science*, 132:937–946, 2018.

[7] Shufeng Xiong, Hailian Lv, Weiting Zhao, and Donghong Ji. Towards twitter sentiment classification by multi-level sentiment-enriched word embeddings. *Neurocomputing*, 275:2459–2466, 2018.

[8] G. Mishne and N. Glance. Predicting movie sales from blogger sentiments. In *Proc. of Computational Approaches to Analyzing Weblogs, Papers from the 2006 AAAI Spring Symposium*, pages 1–4, Stanford, California, USA, 2006.

[9] O'Connor, R. Balasubramanyan, B. Routledge, and N. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proc. of AAAI Conf. on Weblogs and Social Media*, pages 122–129, Stanford, California, USA, 2010.

[10] B. Liu. *Sentiment analysis: mining opinions, sentiments, and emotions*. Cambridge University Press, 2016.

[11] B. Schuller and A. Batliner. *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. Wiley, 2013.

[12] B. Pang and L. Lee. *Opinion mining and sentiment analysis*. Now Publishers, 2008.

[13] A. Kumar and T. Sebastian. Sentiment Analysis: A Perspective on its Past, Present and Futures. *International Journal of Intelligent Systems and Applications*, 4(10):1–14, 2012.

[14] B. Schuller and A. Batliner. *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. Wiley, 2013.

[15] B. Liu. *Sentiment analysis and opinion mining. Synthesis digital library of engineering and Computer Science*. Morgan & Claypool, 2012.

[16] W. Medhat, A. Hassan, and H. Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[17] V. Hatzivassiloglou and K.R. McKeown. Predicting the semantic orientation of adjectives. In *Proc. of ACL'98*, pages 174–181, 1998.

[18] Jorge A. Balazs and Juan D. Velásquez. Opinion mining and information fusion: A survey. *Information Fusion*, 27:95–110, 2016.

[19] F.A. Pozzi, E. Fersini, E. Messina, and B. Liu. *Sentiment analysis in social networks*. Morgan Kaufmann, 2017.

[20] A. Garg and D. Roth. Understanding probabilistic classifiers. machine learning. In *Proc. of 12th European Conference on Machine Learning (ECML'01)*, pages 179–191, Freiburg, Germany, 2001.

[21] C.C. Aggarwal and C. Zhai. *Mining text data*. Springer Science and Business Media, 2012.

[22] P. Nakov, Z. Kozareva, A. Ritte, S. Rosenthal, V. Stoyanov, and T. Wilson. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Proc. of 7th International Workshop on Semantic Evaluation (SemEval'13)*, pages 312–320, Atlanta, Georgia, USA, 2013.

[23] M.M. Bradley and P.J. Lang. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical report, Center for Research inPsychophysiology, University of Florida, 1999.

[24] F.Å. Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In *Proc. of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, pages 93–98, Crete, Greece, 2011.

[25] F. Bravo-Marquez, M. Mendoza, and B. Poblete. Combining strengths, emotions and polarities for boosting twitter sentiment analysis. In *Proc. of Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, pages 1–9, Chicago, USA, 2013.

[26] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proc. of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 168–177, Seattle, WA, USA, 2004.

[27] S.M. Mohammad and P.D. Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[28] J.R. Landis and G.G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.