Sentiment Analysis using Word-Graphs

John Violos
Institute of Communication and Computer Systems,
9, Heroon Polytechniou Str,
15773 Athens, Greece
violos@mail.ntua.gr

Konstantinos Tserpes
Harokopio University of Athens
9, Omirou Str., 17778 Tavros, Greece
tserpes@hua.gr

Evangelos Psomakelis
Harokopio University of Athens
9, Omirou Str., 17778 Tavros, Greece
vpsomak@mail.ntua.gr

Konstantinos Psychas
Department of Electrical Engineering
Columbia University
New York, NY 10027 USA
psychas@ee.columbia.edu

Theodora Varvarigou
Institute of Communication and Computer Systems,
9, Heroon Polytechniou Str,
15773 Athens, Greece
dora@telecom.ntua.gr

ABSTRACT

The Word-Graph Sentiment Analysis Method is proposed to identify the sentiment that expressed in a microblog document using the sequence of the words that contains. The sequence of the words can be represented using graphs in which graph similarity metrics and classification algorithms can be applied to produce sentiment predictions. Experiments that were carried out with this method in a Twitter dataset validate the proposed model and allow us to further understand the metrics and the criteria that can be applied in words-graphs to predict the sentiment disposition of short, microblog documents.

CCS Concepts

CCS → Information systems → Information retrieval → Retrieval tasks and goals → Sentiment analysis

Keywords

Sentiment analysis, word graph representation model, graph similarity metrics, vector classification.

1. INTRODUCTION

This paper deals with the perennial problem of predicting sentiment from short pieces of text, commonly generated in microblog sites. The challenge is great if one considers the limitations that the short text poses and the resulting use of the language which does not adhere completely to grammatical and syntactical rules. However, the value of the application counterweights the difficulty of the endeavor allowing the consumer of the service to quickly and automatically evaluate the public's disposition towards various topics. As such, in the recent years sentiment analysis has become an indispensable tool in marketing and other suchlike application domains.

The proposed approach for sentiment analysis revolves around a novel method for modeling the documents, called Word-Graph Sentiment Analysis Method (WSAM). This approach combines the structure of graphs with graph similarity metrics and classification methods in order to automatically infer the sentiment expressed by the author of the document towards a preselected topic. In the WSAM model the words that are contained in a short document are represented as nodes and the vicinity between the words as edges.

Furthermore, the WSAM model is used to represent the sentiment polarities classes based on a training dataset of annotated documents. Afterwards a graph similarity technique is applied to compare each document graph with the three sentiment polarity graphs. The outcomes of the comparison comprise a new representation of the documents as vectors. Finally the sentiment prediction is carried out using a vector classification method.

Each of the above mentioned steps is described and analyzed in what follows. Many alternatives for each step are examined in order to find the setup that produces the most accurate predictions. The appropriateness and the applicability of the method are demonstrated through metrics that quantify the sentiment similarity that is expressed based on the adjacency of the words.

Experiments are carried out for each of the resulting setups. In addition further experiments carry out with other state of the art methods in the same dataset. The comparison of the results verify the proposed model and the assumption that sentiment is largely hidden not only in the words that one uses (like in all bag-of-words or lexicon-based solutions) but also in their position in the document and their vicinity.

2. RELATED WORK

Sentiment analysis is essentially a two or three class classification task, depending on whether the "neutral" class is considered along with the "positive" and "negative" class or whether general sentiment polarity vs neutrality is examined. Each class represents the disposition of the authors towards a subject as it is expressed by their corresponding documents. Often the problem is tackled as a multi-label classification problem (e.g. [3, 4, 11, 17, 28, 35]) in which the documents are assigned weights indicating the degree of correlation with the corresponding class. Of course, this is the case in the majority of the literature works; there is also a large body of scientific publications (e.g. [18, 19, 23]) and even tools ([27, 32]) that deal with the problem of sentiment analysis as a continuous and multi-dimensional sentiment problem, searching in documents for multiple types of sentiment rather than just three.

WIMS '16, June 13-15, 2016, Nîmes, France
© 2016 ACM. ISBN 978-1-4503-4056-4/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2912845.2912863

The key challenge in these kinds of problems is to distinguish the features that stem from the document and potentially from its metadata that will enable the modeling of the document for the purposes of the classification. In turn, this model comprises the input of a function (classifier) which maps the features to an accurate answer to the question: which sentiment does a particular author expresses, about a particular subject, in a particular document. The classifier function is rarely known (if ever). In many cases the classifier is actually a similarity function between the model of the document in question and a known representative model of a class [7]. To infer whether the document belongs or not to a class, the outcome of the similarity function is then compared to a threshold which has been experimentally defined. Machine learning techniques (Naïve Bayes, C4.5, SVM, etc) are commonly employed for determining these thresholds. However, an even more common approach is to use the same techniques for estimating the similarity function altogether.

Regarding the features these are typically extracted by a natural language processing (NLP) technique like ngrams [34] or otherwise by a lexicon ([14, 31]) explicitly devised for supporting sentiment classification and opinion mining applications like SentiWordNet [5]. Another approach that is widely used in sentiment analysis tasks is the "bag of words" and "bag of ngrams" approaches e.g. [25] in which each document is essentially modeled by a set of words or ngrams respectively. An alternative approach is the use of features that are contextually related to the document (e.g. [1, 2, 4]) especially when sentiment analysis is applied on social media which are rich in metadata and other activities of social context (e.g. "mentions", "likes", etc).

There are also models that are not NLP as the recently emerged ontology-based approaches [12]. Ontology is defined as a set of representational primitives and can model a domain of knowledge. Afterwards the sentiment analysis carried out based on the concepts and the properties that included in the ontology [16] or using a query analysis engine module that analyze different extracted words [30]. These words are mapped to concepts, attributes and ontology instances and are used for an ontology search that relates the emotional difference between a query and a text.

Among the various NLP models, one could distinguish the word2vec model [21] which either in its skip-gram or in its continuous bag of words (CBOW) form considers the context of a word, i.e. the words that will most likely follow or precede the word in question in a document. A similar approach is the n-gram graph model ([2, 3]) which employs a graph representation of adjacency and adjacency frequency of ngrams in a document. We study a variation of the latter model creating the graph of words rather than ngrams. This model is easier to be created than the word2vec model and might be more appropriate for capturing neologisms, abbreviations and common grammatical errors [3].

3. PROPOSED MODEL

The method is inspired by previous work, i.e. the use of ngram-graphs for single-label classification tasks [9] and especially the work presented in [33]. The differentiation lies in that we propose the creation of word-graphs as opposed to ngram-graphs with the intention to better capture the encompassed notions of sentiment in a specific document for a specific subject.

We propose a graph theoretic and vector classification aproach to conduct sentiment analysis in short documents that can be produced from microblog users. Each such document can be represented as a directed unweighted graph based on the terms that it contacts as well as their vicinity. The sentiment dispositions (positive, negative and neutral) can also be represented as directed unweighted graphs based on annotated training data. The comparison of the graph of the microblog with the three graphs of the sentiment dispositions yields a three dimensional vector. Afterwards, the original text is assigned a sentimental polarity using a classifier of vectors. Each of these steps will be described in the following paragraphs.

Each new document that we want to assess its disposition towards the sentiment classes (i.e. classify it) is transformed to a graph and then we measure its "similarity" to each of the gold standards. Similarity is measured in multiple ways and this is one more novelty of this work: Typical approaches like containment, value and normalized value similarity are used to compare graphs in the context of NLPs and machine learning. This work also employs metrics based on the maximum common subgraph similarity with three different variations to identify the maximum common parts between the graphs and evaluate the quality of the resulting graph.

3.1 Graph Construction

The construction of the word-graphs is based on the words that are contained in a document and their vicinity. The process is similar for documents and sentiment polarities. Each word that is contained in the original text is represented by a labeled node. Two labeled nodes are joined by an edge if their corresponding words are close in the original text. The edges are directed to capture the sequence of the words as they exist in the original documents.

The closeness between the words is represented by the edges that connect the nodes and it is defined by a specific number of words that are following the target word. A frame of words slides across the text and designates the nodes and the edges of the graph as it is illustrated in Figure 1. The size of the frame is a parameter that largely affects the accuracy of the method. As it will be shown in the evaluation section we have conducted experiments with a frame size ranging from 2 to 10. Microblog documents commonly contain a small amount of words either because there is a limitation in the amount of characters (e.g. Twitter) but also because of the notion of brevity that they are encompassing. This implies that there is no point in investigating a frame size larger than 10.

It is important to state that the nodes of the graph can be any sequence of characters that comprise the tokens in a tokenization process where the tokenizer is a blank space. We commonly refer to these tokens as words, even though they might not be so, especially in microblogging where otherwise meaningless character sequences express a meaning for their writers. In the proposed mode, these sequences of characters also comprise labeled nodes in the graph.

Regarding the 3 sentiment classes graphs, each of them is constructed by merging all the graphs that model the individual short documents that belong to the corresponding class. That is, all document graphs that fall in a single category are merged to eventually comprise the sentiment class representative graph. The merging is straightforward: common edges and nodes are kept once in the new graph and every non-common element is added in reference to the common ones.

3.2 Graph Similarity

The graph similarity between the graph of a document and the graph of a sentiment class can indicate the degree that a document expresses the corresponding sentiment. There is a plethora of

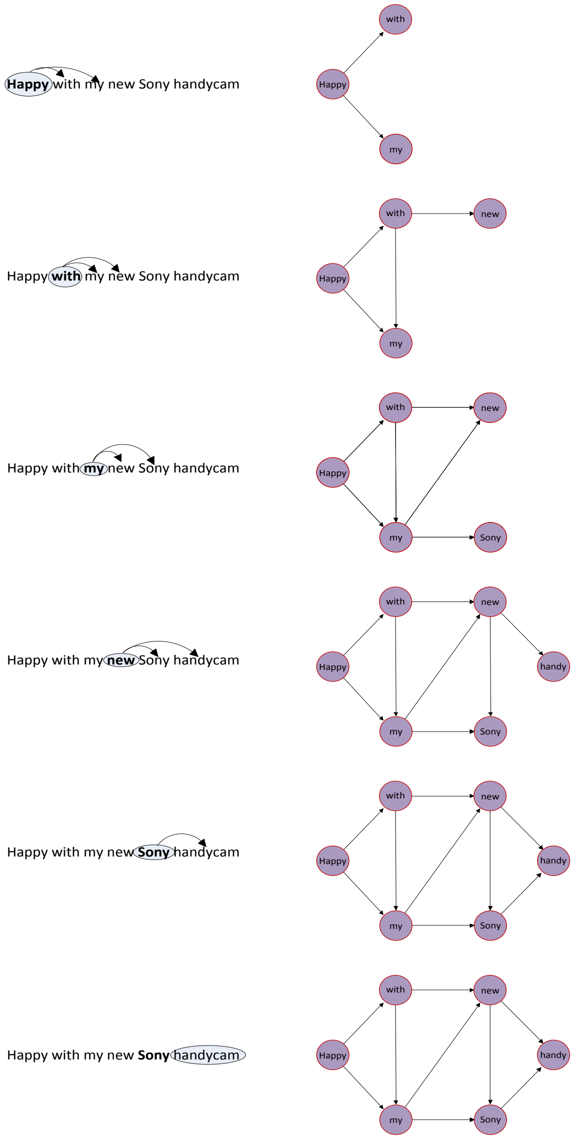
Happy with my new Sony handycam
Happy
with
my
Happy with my new Sony handycam
with
new
Happy
my
Happy with my new Sony handycam
with
new
Happy
my
Sony
Happy with my new Sony handycam
with
new
Happy
handy
my
Sony
Happy with my new Sony handycam
with
new
Happy
handy
my
Sony
Happy with my new Sony handycam
with
new
Happy
handy
my
Sony

Figure 1. Graph construction based on a microblog short document.

Figure 1 illustrates how the frame slides toward the end of the document; in this example the frame is the two following words and for each word some edges and nodes are added to the graph.

methods to estimate the graph similarity. Two typical approaches are to recognize the maximum common subgraph [29] and to match neighboring nodes[24]. For the purposes of our study we need a graph similarity method that uses as parameters the labels of the nodes, the direction of the edges and the number of the common edges.

Figure 2 illustrates that a graph of a document can be compared with the graph of each sentiment class in order to produce three numbers. Each number quantifies the correlation of the target document to the three sentiment classes. These three numbers comprise the vector that is used in a classifier in order to be predicted the most appropriate sentiment. In what follows we provide an account of the similarity metrics we used in our approach.

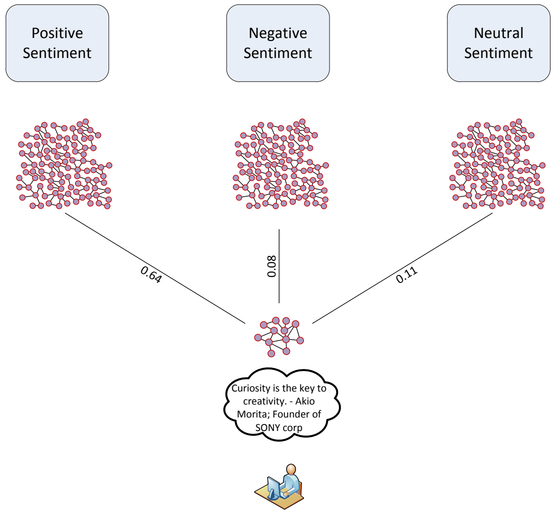
Positive Sentiment
Negative Sentiment
Neutral Sentiment
0.64
0.08
0.11
Curiosity is the key to creativity. - Akio Morita; Founder of SONY corp

Figure 2. Graph comparison between the word-graph of a document with the word-graphs of the Sentiment Polarities.

Containment Similarity (CS): The CS is a graph similarity measure that has been used in the gauging of the similarity between N-gram graphs [2]. This similarity measure expresses the number of common edges between the two graphs by the number of the edges of the smaller graph.

$$\boldsymbol{CS(G_T,G_S)} = \frac{\boldsymbol{\sum_{e\in G_T}\mu(e,G_S)}}{\boldsymbol{min(|G_T|,|G_S|)}}\ (1)$$

Where G_T is the word graph of a document, G_S is the word graph of a sentiment polarity $|G_T|$ and $|G_S|$ are their corresponding sizes. The graph size can be the amount of nodes or edges that are contained. e is an edge of a word graph.

Maximum Common Subgraph (MCS)-based similarity metrics: We have also explored similarity measures that are based on the size of the MCS between two graphs. The detection of the MCS in unlabeled graphs is a NP-complete problem [8]. Fortunately, the detection of MCS in graphs with labeled nodes is a process of linear complexity. Three variations of the metric that use the MCS are described in the equation 2, 3 and 4.

$$MCSNS = \frac{\boldsymbol{MCSN(\,|G_T|,|G_S|)}}{\boldsymbol{min(|G_T|,|G_S|)}}\ (2)$$

where $MCSN(\,|G_T|,|G_S|)$ is the total number of nodes that are contained in the MCS of G_T and G_S graphs.

$$MCSUES = \frac{\boldsymbol{MCSUE(\,|G_T|,|G_S|)}}{\boldsymbol{min(|G_T|,|G_S|)}}\ (3)$$

where $MCSUE(\,|G_T|,|G_S|)$ is the total number of the edges contained in the MCS regardless the direction of them in the graphs G_T and G_S

$$MCSDES = \frac{\boldsymbol{MCSDE(\,|G_T|,|G_S|)}}{\boldsymbol{min(|G_T|,|G_S|)}}\ (4)$$

where $MCSDE\ (|G_T|\,,|G_S|)$ is the number of the edges contained in the MCS and have the same direction in the graphs G_T and G_S

The metrics MCSNS, MCSUES and MCSDES are based on the MCS between the two graphs but each of them quantify the graph similarity in a different way. MCSNS uses the amount of nodes that exist in MCS. In a more thorough investigation this means that it considers the amount of different words which are related to one another in the original document and the documents of the corresponding sentiment class.

The metrics MCSUES and MCSDES use the amount of edges that exist in the MCS. This implies that we are interested in the amount of edges that exist in the MCS. In the case of MCSDES we are taking into account the direction of the edges whereas in the case of MCSUES direction is not considered. Further investigation reveals that these metrics capture the notion of how strong is the relation of the words that coexist in the document and the sentiment class.

3.3 Classification

The proposed sentiment analysis method splits the training data of documents in two parts. The first part is used to make the word graph representation for each sentiment class. The second part of documents is used to train a vector classifier such as SVM and Bayes.

Each document of the second part of the training dataset is represented with a word graph. Afterwards these documents' word graphs are compared with three sentiment classes' graphs using one of the graph similarity measures that are described in the Section 3.2. The comparison results form a three-dimensional vector. This vector represents the document's polarity. The vectors of the second part of the training dataset are used to train the classifier.

A new target document is again represented as a word-graph and compared with the three sentiment classes' word-graphs in a similar way with the documents of the second part of the training dataset. This generates the vector which is classified by the trained classifier in one of the three classes that represent each sentiment class.

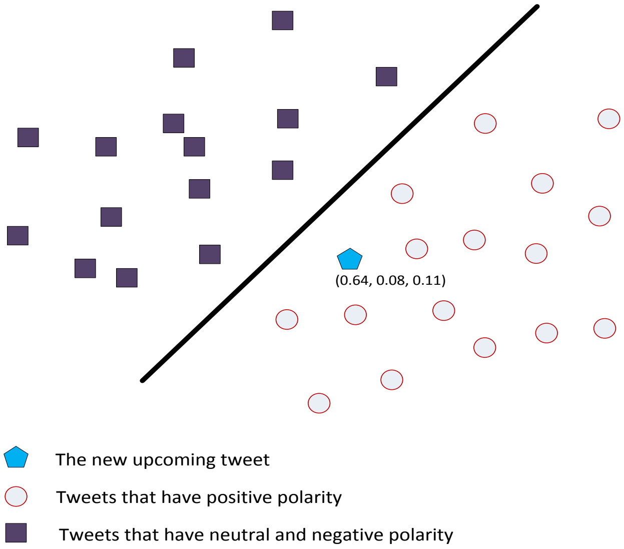
(0.64, 0.08, 0.11)
The new upcoming tweet
Tweets that have positive polarity
Tweets that have neutral and negative polarity

Figure 3.Classification of a new target document using its vector representation.

Two different classification methods the Support Vector Machine (SVM) and Bayes Classification were investigated for the classification of the document vectors. The SVM model represents the instances as points in an N-dimensional space. Afterwards a planar should be gauged to divide the instances that belong to different categories as it is illustrated in Figure 3. The gap between instances that belong to different categories should be as wide as possible. SVM was developed initially for binary classification problems. The sentiment analysis problem requires three classes so a variation of the SVM for multi-class classification [26] was used. In particular a linear SVC was selected with one-vs.-one classification [10]. In one-vs.-one classification a separated binary classifier is trained for each pair of labels. A new target document is applied to all binary classifiers and finally the class that yields the highest number of predictions will be the class in which the document will be assigned.

Gaussian Bayes classifier [13] uses a conditional probability model in which a document can be represented as a vector as described in the previous paragraphs. The three values of the vector $S_i(s_+, s_-, s_=)$ are the three independent variables and the sentiment position is the dependent variable y. The likelihood of a document to express a sentiment is assumed to be as given in the equation 5

$$P(S_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad (5)$$

Where σ_y and μ_y are estimated using maximum likelihood and the values s_+, s_-, and $s_=$ from the vector S_i are estimated by the word graph of the document.

3.4 Graph Filtering

Many machine learning techniques use feature selection criteria in order to make dimensionality reduction and improve the methods accuracy [6]. The feature selection criteria filter out terms such as words, n-grams or numbers that do not have a positive contribution in the method's process. In our study we will experiment with a well-known feature criterion called Mutual Information [36] using as terms the edges of the graphs.

The mutual information criterion will be applied after the word graph creation of the training documents and sentiment classes in order to discard an amount of edges from the graphs of sentiment position. The set of the edges that are discarded are discarded by the new target documents.

The benefits of using the mutual information are that the graphs become smaller so the sentiment analysis method has less demands for memory and computational resources. Furthermore, there are cases in which other machine learning methods presented an improved accuracy. Unfortunately, as we will see and explain in the experimental section, this does not occur with WSAM.

The edge filter out criteria is used only with the containment similarity metric and not with the metrics that are based on the MCS because the potential discarded edges were often contained in the MCS and removing them affects negatively the accuracy of the method.

The mutual information between an edge with the graph of a sentiment class is defined by the equation 5

$$I(e, G_S) = log\,\frac{A \cdot N}{(A+C) \cdot (A+B)}\ (5)$$

where:

A is the number of times that the edge e exists in the graphs of the documents from the second part of the training data and in the graph of the sentiment class.

B is the number of times that the edge e exists in the graph of the documents from the second part of the training data but does not exist in the graph of the sentiment class.

C is the number of times the second part of the training documents express the sentiment of the G_S but do not contain the edge e.

N is the total number of the documents from the second part of the training data.

To measure the positive contribution of an edge in the class prediction process, we calculate the factor $I(e, G_S)$ for positive S^+ , negative S^- and neutral $S^=$ sentiment classes. Finally, equation 6 is used to calculate the global mutual information for each edge.

$$I_{avg}(e) = \sum_{i\epsilon(+,-,=)} P_r(S^i) \cdot I(e, G_i)\ (6)$$

where $P_r(S^i)$ is the percentage of the documents from the second part of the training dataset that express the sentiment S^i.

The $I_{avg}(e)$ expresses the appropriateness and the contribution of each edge for the accurately sentiment prediction in the WSAM. We can set a threshold for $I_{avg}(e)$ and discard all the edges that are smaller than this.

3.5 Overview of the Word Graph Sentiment Analysis Method

In the previous sections we described the construction of the word-graphs for documents and sentiment classes, the graph similarity metrics, the vector representation of document sentiment and the construction of sentiment classifiers. This section summarizes all of them with an example. It explains the whole method and describes how the sentiment of a new target document can be identified. Figure 4 illustrates all the stages of the word graph sentiment analysis method.

Stage 1

An annotated dataset of documents is needed to train WSAM. The ground truth includes the document text and its corresponding sentiment class.

Stage 2

The dataset is divided in two equal parts which are used in a different way. In the 1st part, the documents that express the same sentiment are grouped together.

Stage 3

The groups of the tweets that belong to the same sentiment are used to construct the word graphs for the three sentiment classes. Each class graph is constructed from the documents that express the corresponding sentiment disposition. Each document of the 2nd part is represented as a word graph too.

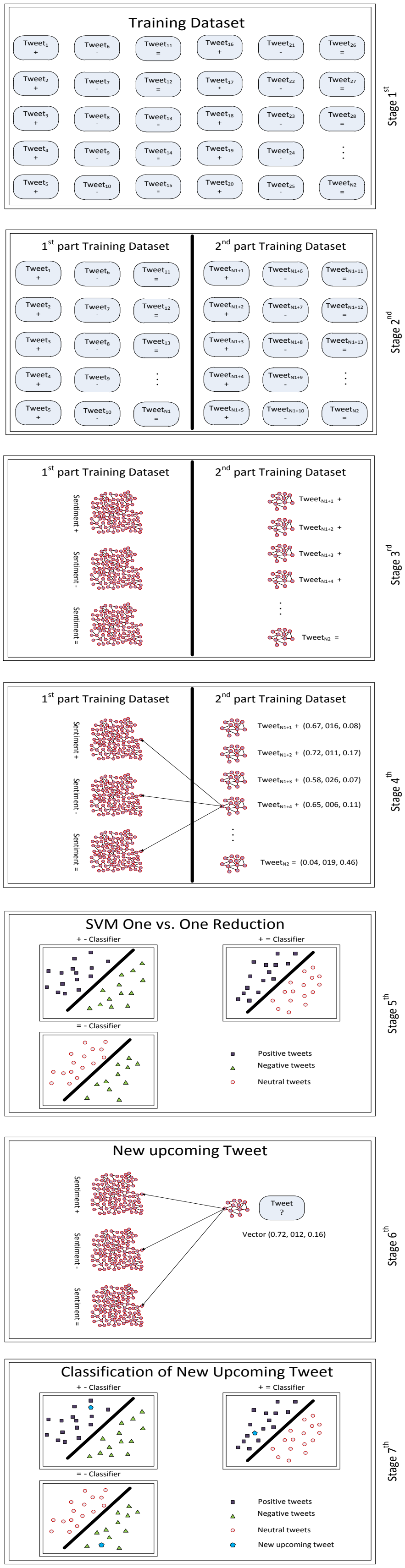
Training Dataset
Tweet1 + Tweet6 - Tweet11 = Tweet16 + Tweet21 - Tweet26 =
Tweet2 + Tweet7 - Tweet12 = Tweet17 + Tweet22 - Tweet27 =
Tweet3 + Tweet8 - Tweet13 = Tweet18 + Tweet23 - Tweet28 =
Tweet4 + Tweet9 - Tweet14 = Tweet19 + Tweet24 - ⋮
Tweet5 + Tweet10 - Tweet15 = Tweet20 + Tweet25 - TweetN2 =
Stage 1st
1st part Training Dataset
2nd part Training Dataset
Tweet1 + Tweet6 - Tweet11 = TweetN1+1 + TweetN1+6 - TweetN1+11 =
Tweet2 + Tweet7 - Tweet12 = TweetN1+2 + TweetN1+7 - TweetN1+12 =
Tweet3 + Tweet8 - Tweet13 = TweetN1+3 + TweetN1+8 - TweetN1+13 =
Tweet4 + Tweet9 - ⋮ TweetN1+4 + TweetN1+9 - ⋮
Tweet5 + Tweet10 - TweetN1 = TweetN1+5 + TweetN1+10 - TweetN2 =
Stage 2nd
1st part Training Dataset
2nd part Training Dataset
Sentiment +
Sentiment -
Sentiment =
TweetN1+1 +
TweetN1+2 +
TweetN1+3 +
TweetN1+4 +
TweetN2 =
Stage 3rd
1st part Training Dataset
2nd part Training Dataset
Sentiment +
Sentiment -
Sentiment =
TweetN1+1 + (0.67, 016, 0.08)
TweetN1+2 + (0.72, 011, 0.17)
TweetN1+3 + (0.58, 026, 0.07)
TweetN1+4 + (0.65, 006, 0.11)
TweetN2 = (0.04, 019, 0.46)
Stage 4th
SVM One vs. One Reduction
+ - Classifier
+ = Classifier
= - Classifier
Positive tweets
Negative tweets
Neutral tweets
Stage 5th
New upcoming Tweet
Sentiment +
Sentiment -
Sentiment =
Tweet ?
Vector (0.72, 012, 0.16)
Stage 6th
Classification of New Upcoming Tweet
+ - Classifier
+ = Classifier
= - Classifier
Positive tweets
Negative tweets
Neutral tweets
New upcoming tweet
Stage 7th

Figure 4. The stages of the Word Graph Sentiment Analysis Method

Stage 4

The document graphs of the 2nd part are compared with a graph similarity metric with the three word-graphs. This comparison results in three figures. Each figure expresses the similarity of the graph of the document with the graph of the sentiment polarity. From now on, each document of the 2nd part can be represented as a vector of these three numbers.

Stage 5

The vectors from the documents of the 1st part of the training dataset accompanied with the sentiment that they express are used to train an SVM Classifier. In fact, three SVM classifiers are trained because of the need for a multi-class classification. Each class represents the sentiment polarity.

Stage 6

A new target document is represented as a word graph in a similar way with the documents of the 2nd part in stage 3. Afterwards the graph of the new target document is compared with the sentiment classes' graphs and is represented as a vector as we did in stage 4.

Stage 7

The vector of the new target document is mapped as a point in the feature space of the SVM classifiers that we constructed in the 5th stage. The corresponding sentiment prediction is made based on that.

Note that the graph filtering process is not illustrated in Figure 4 though the edge removal part takes place in stage 3 and 6. The overview of the proposed model is described with the SVM classification. The process is similar in the Bayes classification case. For the similarity metric in the stage 4 and 6, we can use any metric from those described in Section 3.2 with the restriction that it would be the same metric in both stages.

The complexity of the method is low. The process of the graph creation is linear. All the characters of a text are iterated from the first to the last and for each character is created the corresponding node and the directed edges that join it with the previous nodes. The comparison between two labeled graphs using maximum common subgraphs criteria has complexity $O(V^2 + V \cdot E)$ where E is the number of edges and V the number of nodes. The complexity of the maximum common subgraph criteria can be reduced to the complexity of the depth-first search or breadth-first multiplied by the number of nodes. The complexity of the containment similarity criterion is $O(E_{G1} \cdot E_{G2})$. Where E_{G1} and E_{G2} are the numbers of the edges of the graph G1 and G2 respectively.

An issue that is always emerged in the supervised machine learning techniques is the need of labeled training dataset. The proposed sentiment analysis model in order to make predictions needs a corresponding training dataset. These datasets should contain posts in the language of the testing texts but in the most cases they are not dependent of a specific domain. The manual annotating process is a tedious work because the texts should be read by people who will label them with subjective criteria.

4. EXPERIMENT SETUP

To conduct the experiments we used the processed version of the public available dataset from the paper of Sascha, Hufenhaus and Albayrak Language-Independent Twitter Sentiment Analysis [22]. This dataset consist of tweets that were selected randomly and tweets that contain a brand name like "sony", "audi", etc.

The dataset annotated manually by workers on Amazon's Mehanical Turk. It contains 10594 tweets. 1486 are negative, 2334 are positive and 6774 are neutral. The average term length of the tweets is 14.2. The dataset also includes three more set of tweets in Portuguese, German and French but the available tweets for these languages range between 2,000 to 2,500. This number of tweets is not sufficient to produce well-founded results for training and testing.

We studied the proposed model as described in Section 3 using the 10-fold cross validation approach [15]. In each fold we used 90% of the tweets for training the sentiment analysis algorithm and 10% for testing. No external textual data were used for training. The work of Sascha, Hufenhaus and Albayrak used a dataset of 800 million tweets making the two works incomparable.

The experiments were conducted in a commodity desktop computer. The Java programming language is used for the implementation and application of the software modules that perform graph construction, graph comparison, feature filtering and the construction of the vector representation of the tweets. The classification of the vectors that represent the tweets and the evaluation of the method occurred using the scikit library in the Python programming language. The experiments' source code (in Java), the python script and the input of the method are available for any kind of reproduction and reexamination in a publicly accessible URL: http://users.ntua.gr/violos/.

To measure the execution time we needed to count in the time for the execution of the ten cross validation training technique, the construction of the three sentiment classes' graphs, the construction of the vector representation and the application of the classifier. In most cases this pipeline lasted less than 30 seconds in a commodity desktop computer.

Our approach was compared with numerous other approaches using the same dataset. In particular, we compared against a set of NLPs and learning algorithms which were employed in [28], the results of which are presented in Table 1.

Method	4Grams	4Gram Graphs
Bayesian Network	0.6788	0.6791
C4.5	0.6828	0.6896
Support Vector Machines	0.6777	0.6847
Logistic Regression	0.6822	0.7115
Simple Logistic Regression	0.6816	0.7109
Multi-Layer Perceptron	0.6788	0.7069
Best-First Tree	0.6790	0.6840
Functional Tree	0.6822	0.7079

Table 1. Sentiment Analysis Accuracy using other state of the art methods in the same dataset.

The highest accuracy achieved with the Logistic Regression (LR) using 4Gram Graphs, i.e. 0.7115. This outcome is used in the following figures as baseline. The Word Graph Sentiment Analysis surpasses it in the most cases.

The Word graph Sentiment analysis method has been investigated with many variations in order to understand the applicability of the word graphs in sentiment analysis and to conclude in a setup that increases the method accuracy. The experiments were carried out with a word frame that spanned from two to ten.

As we mentioned in Section 3 of the proposed model, we regard the investigation of graphic similarity metrics as a very important step. Figure 5 summarizes the results in which all metrics were used: Containment Similarity (CS), Maximum Common Subgraph Node Similarity (MCSN), Maximum Common Undirected Edges Similarity (MCUES) and the Maximum Common Directed Edges Similarity (MCDES).

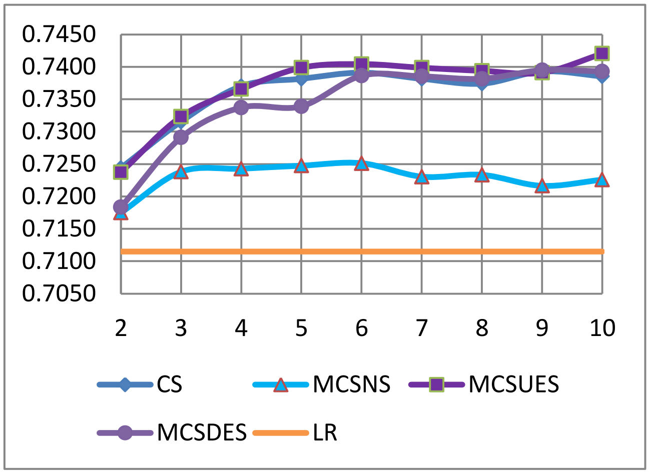

Figure 5 Graph Comparison Metrics Evaluation

The CS is based on the amount of common edges, that is to say the neighborhood between two words. While the methods that use the Maximum Common Subgraph (MCS) exploit better the structure of graphs. The MCSNS is based on the amount of nodes in the MCS. On the other hand MCSUES and MCSDES are based on the amount of edges in the MCS. This verifies our intuition belief that the vicinity between words expresses the sentiment position of a tweet in a better way than the simple existence of the words. The criteria that use the MCS do not express only how many couples of words are close one to the other but they also express that a set of words are related because they are close in the original training texts.

MCSUES and MCSDES have better accuracy than MCSNS. MCSUES and MCSDES count the edges that exist in the MCS between the tweet and the sentiment word-graph. While MCSNS count the nodes that exist between them. Looking deeper of this result we can understand that it is more important the degree of relation between of the represented words in MCS than the amount of represented words.

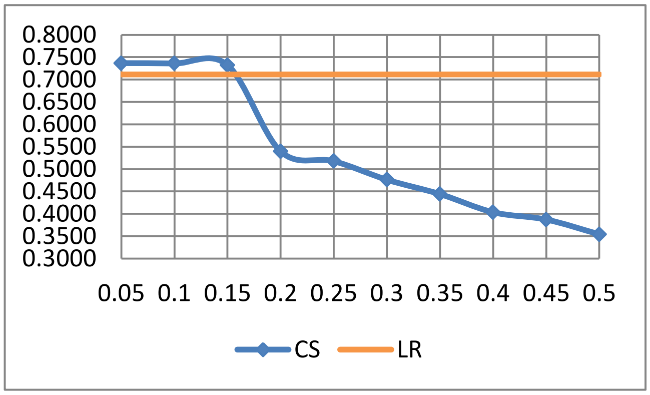

Figure 6 Feature Selection Evaluations

The next experiments that carried out were with the Mutual information (MI) Feature selection criterion in order to filter out an amount of edges. MI cannot be applied in combination with MCSN, MCSUES and MCDES because it often discards the edges that exist in the MCS. The MCS that remains after the filter out criterion is not sufficient to express the similarity of the two graphs because it is very small. In many cases it is constituted only from one or two nodes.

CS can be used in combination with MI because CS is based on the common edges that exist between these two graphs. The experiments that are illustrated in Figure 6 were carried out using CS with fourwords frame. As we can see the more edges are filtered out, the more the accuracy of the method is decreased. The process of filtering out the edges results in smaller graphs and the need in computation but the accuracy is decreased dramatically. The x-axis of Figure 6 represents the MI as described in equation 6. It is important to say that accuracy is decreased even if we discard the edges that have very small MI.

The experiments in Figure 5 were carried out using the SVM classification method to classify the tweet vectors as described in Section 3. In the most cases SVM for classification purposes outperform other classification methods [20]. The word graph sentiment analysis methods was also combined with other classification methods and it is found that using a Gaussian Bayes classifier we can have even better experimental results. The experiments were carried out using MCSUE and CS graph similarity metrics. The accuracy of the method using four-word graphs, MCSUE and Bayes Gaussian Classifier is 75.07%. The reason that SVM shows a lower classifier accuracy than GBC is that it uses a low-dimensional feature space.

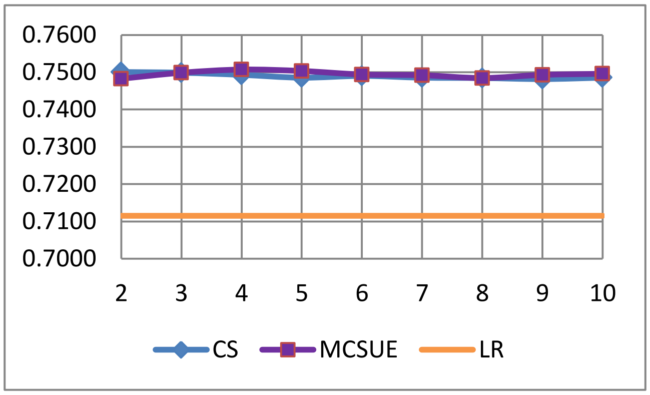

Figure 7 Gaussian Naïve Bayes Classification Evaluation

5. CONCLUSIONS

An innovative sentiment analysis method is proposed that combines the well-defined structure of graphs with classification algorithms. The word graphs can capture the sequence of the words that are contained in a microblog document. Several graph similarity techniques were applied so to estimate the similarity between the document graphs and the graphs that represent the sentiment classes. The result of the comparison is a 3-feature vector which captures the sentiment disposition of the document. The sentiment prediction is conducted exactly by feeding this vector in a classifier. For all the steps of the methods we applied various metrics and methods that are intrinsic to the needs of the needs of our research.

The vicinity and the order between the words are proved to be a good source of information to predict the sentiment of a microblog document and specifically a tweet. We conclude that more advanced graph similarity metrics that gauge the strength of the relation between the most common words (MCSUES) can have better outcomes than graph similarity metrics that are based to their amount of common words that are neighbor (MCSNS) and the amount of the pair of words that are close (CS).

The rank of the word frame that it is used to construct the word graphs affects the accuracy of the method. We noticed that as the rank of the threshold is increased the accuracy in combination with CS, MCSUES, and MCSDES is also increased. On the other hand it or stay stable or it is slightly decreased using a frame bigger than 3 with the MCSNS criterion. The reason is that CS, MCSUES, and MCSDES criterions are based on the amount of the common edges that exist between the two graphs. A high rank of word threshold will catch the relation of words more than a low rank. The MCSNS is based on the common nodes that exist between the two graphs, so it is not affected by the rank of word threshold.

Two more things that were studied in our research is the classification of the vectors that represent the tweets and the combination of the method with a feature selection criterion. The word graph sentiment analysis method should be used with a classification method that makes good predictions in a low dimensional space. The Gaussian Bayes classifier can be trained easier and produce more accurate predictions than other classifiers.

A feature selection method can be used to decrease the amount of the data that need to be processed. We made experiments using the Mutual Information metric but it is noticed that the accuracy of the method was decreased dramatically as the amount of the edges that removed was increased.

The experimental results show that the proposed model is practical, effective and in most of the cases it outperforms other state of the art sentiment analysis methods. The word graph representation of the tweets and the sentiment positions is a data structure that captures the sentimental information of a tweet and using a graph similarity metric and a classification method it is can be predicted.

Further research will be carried out in order to be improved the proposed model. It is presented a model that uses unweighted graphs. We also plan to represent texts as weighted words graphs. The weights can indicate the frequency that two words coexist. In addition the graph similarity metrics that have been used are based more on the common edges. We plan to investigate alternative graph similarity metrics that takes into account both edges and nodes. The common edges and nodes will be multiplied with two coefficients in order to detect in what degree the words and the vicinity of the words can express the sentiment of a text.

6. ACKNOWLEDGMENTS

This work has been supported by the SUPER and CONSENSUS projects and has been partly funded by the EU Seventh Framework Programme, themes: SEC-2013.6.1-1: The impact of social media in emergencies – Capability Project under Contract No. 606853 and ICT-2013.5.4: ICT for Governance and Policy Modelling under Contract No. 611688

7. REFERENCES

[1] Agarwal, B. et al. 2015. Sentiment analysis using common-sense and context information. *Computational intelligence and neuroscience*. 2015, (2015), 30.

[2] Aisopos, F. et al. 2012. Content vs. Context for Sentiment Analysis: A Comparative Analysis over Microblogs. *Proceedings of the 23rd ACM Conference on Hypertext and Social Media* (New York, NY, USA, 2012), 187–196.

[3] Aisopos, F. et al. 2011. Sentiment analysis of social media content using N-Gram graphs. *Proceedings of the 3rd ACM SIGMM international workshop on Social media* (2011), 9–14.

[4] Aisopos, F. et al. 2012. Textual and contextual patterns for sentiment analysis over microblogs. *Proceedings of the 21st international conference companion on World Wide Web* (2012), 453–454.

[5] Baccianella, S. et al. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. (May 2010).

[6] Chagheri, S. et al. 2012. Feature vector construction combining structure and content for document classification. *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)* (Mar. 2012), 946–950.

[7] Chen, Y. et al. 2009. Similarity-based Classification: Concepts and Algorithms. *J. Mach. Learn. Res.* 10, (Jun. 2009), 747–776.

[8] Conte, D. et al. *Challenging Complexity of Maximum Common Subgraph Detection Algorithms: A Performance Analysis of Three Algorithms on a Wide Database of Graphs*.

[9] Giannakopoulos, G. et al. 2008. Summarization System Evaluation Revisited: N-gram Graphs. *ACM Trans. Speech Lang. Process.* 5, 3 (Oct. 2008), 5:1–5:39.

[10] Gunn, S.R. 1998. *Support Vector Machines for Classification and Regression*.

[11] Huang, S. et al. 2013. Sentiment and Topic Analysis on Social Media: A Multi-task Multi-label Classification Approach. *Proceedings of the 5th Annual ACM Web Science Conference* (New York, NY, USA, 2013), 172–181.

[12] IEEE Xplore Abstract - An ontology based sentiment analysis for mobile products using tweets: *http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6921974&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6921974*. Accessed: 2016-04-13.

[13] John, G. and Langley, P. 1995. Estimating Continuous Distributions in Bayesian Classifiers. *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (1995), 338–345.

[14] Khan, A.Z. et al. 2015. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*. (2015), 89.

[15] Kohavi, R. 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (San Francisco, CA, USA, 1995), 1137–1143.

[16] Kontopoulos, E. et al. 2013. Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*. 40, 10 (Aug. 2013), 4065–4074.

[17] Liu, S.M. and Chen, J.-H. 2015. A Multi-label Classification Based Approach for Sentiment Classification. *Expert Syst. Appl.* 42, 3 (Feb. 2015), 1083–1093.

[18] Maas, A.L. et al. 2011. Learning Word Vectors for Sentiment Analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (Stroudsburg, PA, USA, 2011), 142–150.

[19] Maas, A.L. et al. *Multi-Dimensional Sentiment Analysis with Learned Representations*.

[20] Maglogiannis, I.G. 2007. *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*. IOS Press.

[21] Mikolov, T. et al. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* (2013), 3111–3119.

[22] Narr, S. et al. 2012. Language-independent twitter sentiment analysis. *Knowledge Discovery and Machine Learning (KDML), LWA*. (2012), 12–14.

[23] Neviarouskaya, A. et al. 2010. Recognition of Affect, Judgment, and Appreciation in Text. *Proceedings of the 23rd International Conference on Computational Linguistics* (Stroudsburg, PA, USA, 2010), 806–814.

[24] Nikolić, M. 2012. Measuring similarity of graph nodes by neighbor matching. *Intelligent Data Analysis*. 16, 6 (Jan. 2012), 865–878.

[25] Pak, A. and Paroubek, P. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta* (2010).

[26] Pal, M. 2008. Multiclass Approaches for Support Vector Machine Based Land Cover Classification. *arXiv:0802.2411 [cs]*. (Feb. 2008).

[27] Polymerou, E. et al. 2014. EmoTube: A Sentiment Analysis Integrated Environment for Social Web Content. *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)* (New York, NY, USA, 2014), 20:1–20:6.

[28] Psomakelis, E. et al. 2014. Comparing methods for Twitter Sentiment Analysis. *arXiv:1505.02973 [cs]* (2014).

[29] Raymond, J.W. and Willett, P. 2002. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*. 16, 7 (Jul. 2002), 521–533.

[30] Sam, K. M and Chatwin, C. R. Ontology-Based Sentiment Analysis Model of Customer Reviews for Electronic Products. *International Journal of e-Education, e-Business, e-Management and e-Learning*.

[31] Taboada, M. et al. 2011. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*. 37, 2 (Apr. 2011), 267–307.

[32] Vakali, A. and Kafetsios, K. *Emotion aware clustering analysis as a tool for Web 2.0 communities detection: Implications for curriculum development*.

[33] Violos, J. et al. 2014. Clustering Documents Using the 3-Gram Graph Representation Model. *Proceedings of the 18th Panhellenic Conference on Informatics* (New York, NY, USA, 2014), 29:1–29:5.

[34] Wang, X. et al. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on* (2007), 697–702.

[35] Wilson, T. et al. 2005. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (Stroudsburg, PA, USA, 2005), 347–354.

[36] Xu, Y. et al. 2007. A study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems*. 3, 3 (2007), 1007–1012.