Sentiment Analysis on Movies Reviews Final Project Report

Georgios Dimitropoulos
4727657
TU Delft University
G.Dimitropoulos-1@student.tudelft.nl

Emmeleia Mastoropoulou
4743539
TU Delft University
E.P.Mastoropoulou@student.tudelft.nl

Konstantinos Touloumis
4620666
TU Delft University
k.touloumis@student.tudelft.nl

ABSTRACT

Sentiment analysis is one of the most popular fields for prediction and classification. In this project, two approaches for sentiment analysis of movie reviews are proposed, implemented and evaluated. Sentences on movies reviews contain independent clauses that express different sentiments toward different aspects of a movie. A dataset of a movie is created from the movie review website Rotten Tomatoes. This project tackles the problem how to label reviews related to movies on a scale of five values. The goal of the project is to familiarize with two totally different approaches. Machine learning algorithms and Sentiment Lexicons approach to predict the sentiment. Finally, we summarize the observations, the results and their interpretations.

KEYWORDS

Sentiment analysis, movie review, Naive Bayes Algorithm, SVM Algorithm, Natural Language Processing, Part Of Speech Sentiment Lexicons, kNN Algorithm

INTRODUCTION

Sentiment analysis is becoming one of the most profound research areas for prediction and classification. Every manufacturer, service provider wants to know how much a customer likes their product or service. Thus, sentiment analysis becomes important for businesses to draw a general opinion about their products and services. In our case to formulate a public opinion about a movie. Natural language processing and machine learning techniques made it possible to analyze user reviews and identify the user's opinions towards them. We will model sentiment from movie reviews and try to find out how this sentiment matches with the success for these movies. In more details, if a movie review is positive, somewhat positive, negative, somewhat negative or neutral. Of course, the task of categorizing each sentence as negative, somewhat negative, somewhat positive, positive or neutral can be difficult and tricky and somewhat ambiguous. Sometimes, the task of identifying the exact sentiment is not so clear for an algorithm. Sentiment analysis seems to require more understanding than the usual topic-based classification. For these reasons we try to analyze the problem to obtain a better understanding of how difficult it is.

In this report our goal is the rating-inference problem. More specifically, we are going to tackle the problem how to label reviews related to movies on a scale of five values (5-class classification problem). Different type of classification algorithms have been implemented, namely Naive Bayes, SVMs, Logistic Regression, Neural Networks, 3-NN and Decision trees. In order to calculate the sentiment score of the review, each piece of text can be examined separately or in combination with others. Feature selection is a very important and useful method. We can choose every word, n subsequent words, sentence, and/or the whole review to be represented as a feature. We try to find answers to the following questions: How difficult is the task of extracting sentiment from short comments or sentences ? What machine learning techniques or lexicon based techniques are useful for this purpose? Which one of them performs the best and which techniques are better?

First of all we analyze the current state-of-the-art for sentiment analysis in order to gain a deeper understanding of the problem as well as the methods used to approach it. After that, we present the database and the preprocessing techniques we applied on this database. Thereafter, we present the two totally different approaches we have investigated and implemented, namely one Machine Learning and one Sentiment Lexicons approach. Finally for each approach we mention the experimental results and its interpretations. We finish with discussion and conclusion in last Section.

BACKGROUND

As mentioned in [1] sentiment analysis is the automated extraction of writer's attitude from the text, and is one of the major challenges in natural language processing.

Sentiment analysis can be beneficial in many ways, taking businesses for example, it provides insight by giving them immediate feedback on products, and measuring the impact of their social marketing strategies or deciding the outcome of an advertising campaign. It can also be highly applicable in political campaigns, or any other platform that concerns public opinion. It even has applications to stock markets and algorithmic trading engines.

Sentiment analysis on Movie reviews is one of the most popular fields to analyze public sentiment. In this section some of these researches and their remarks are mentioned.

In [1] it is mentioned that traditional approaches involve building a lexicon of words with positive and negative polarities, and identifying the attitude of the author by comparing words in the text with the lexicon.

Recently, deep learning algorithms have shown impressive performance in natural language processing applications including sentiment analysis across multiple datasets. These models do not need to be provided with pre-defined features hand-picked by an engineer, but they can learn sophisticated features from the dataset by themselves.

The performance of deep learning algorithms is examined in [1]. Firstly some statistical properties of data are explored. The Naive Bayes algorithm is used as a base classifier and then different deep learning algorithms are applied, like Recurrent Recursive and Convolutional Neural Networks. Their results are compared to those of the Naive Bayes.

In [2] it is mentioned that sentiment anaysis of text can be done in 3 ways.

- using a machine learning based text classifier -such as Naive Bayes, SVM or kNN- with suitable feature selection scheme
- using the unsupervised semantic orientation scheme of extracting relevant n-grams of the text and then labeling them either as positive or negative and consequentially the document;
- using the SentiWordNet based publicly available library that provides positive, negative and neutral scores for words

In general, the baseline algorithm for classifying movie reviews consists of tokenization of the text, feature extraction, and classification using different classifiers such as Naive Bayes, MaxEnt, or SVM. The features can be extracted in a Supervised and semi-supervised approaches for building high quality lexicons have been explored in the literature. However, traditional approaches are lacking in face of structural and cultural subtleties in the written language.

In [2] a SentiwordNet based scheme for document-level and aspect-level sentiment classification is explored. The document-level classification involves use of different linguistic features (ranging from Adverb+Adjective combination to Adverb+Adjective+Verb combination). Also a new domain specific feature is devised for aspect-level sentiment classification of movie reviews. This scheme locates the opinionated text around the desired feature in a review and computes its sentiment orientation. The sentiment scores on a particular aspect from all the reviews are then aggregated. This process is carried out for all aspects under consideration. Finally a summarized sentiment profile of the movie on all aspects is presented in an easy to visualize and understandable pictorial form.

In [3] it is mentioned that SentiWordNet is one of these lexicons that assigns to each synset of WordNet three sentiment numerical scores, positivity, negativity and objectivity. Using opinion lexicons in opinion mining research stems from the hypothesis that individual words can be considered as a unit of opinion information, and therefore may provide clues to review sentiment. This paper presents the results of applying the SentiWordNet lexical resource using various techniques to the problem of automatic sentiment classification of movie reviews.

Similarly in [4] the goal is to study how public mood influences the overall movie review. The researchers calculate the sentiment of each sentence using word stem tokenization. The sentences once split in the form of tokens are compared with an exhaustive positive, negative word dictionary. A sentiment value is calculated for each sentence and is classified in a sentiment class based on the majority of positive, negative or neutral words. A sentiment can be positive, neutral or negative. The sentiment of the movie review is calculated by the auto summation of the sentiment values. The results are quite encouraging and can be nearly accurate by averaging all sentiments.

In [5] a meta-algorithm is applied, based on a metric labeling formulation of the problem, that alters a given n-ary classifier's output in an explicit attempt to ensure that similar items receive similar labels. It is shown that the meta-algorithm can provide significant improvements over both multi-class and regression versions of SVMs when we employ a novel similarity measure appropriate to the problem.

In [6], the goal of the research is not to create or choose an appropriate sentiment lexicon, but rather it is to discover useful term features other than the sentiment properties. For this reason SentiWordNet, is utilized throughout the experiment. A model is proposed of term weighting into a sentiment analysis system utilizing collection statistics, contextual and topic related characteristics as well as opinion related properties. The method proposed above contradicts the lexicon based methods proposed above.

Similarly, in [7] a more complex method is proposed. In this paper, the problem of review mining and summarization is decomposed into the following subtasks: 1) identifying feature words and opinion words in a sentence; 2) determining the class of feature word and the polarity of opinion word 3) for each feature word, first identifying the relevant opinion words, and then obtaining some valid feature opinion pairs 4) producing a summary using the discovered information. A multi-knowledge based approach is proposed to perform these tasks. First, WordNet, movie casts and labelled training data were used to generate a keyword list for finding features and opinions. Then grammatical rules between feature words and opinion words were applied to identify the valid feature-opinion pairs. Finally, the sentences are reorganized according to the extracted feature-opinion pairs to generate the summary.

APPROACH

A) Architecture

Our dataset is a Rotten Tomatoes movie review dataset which is a corpus of movie reviews used for sentiment analysis. It was collected by Pang and Lee [5]. In this work we try to implement our sentiment-analysis approach on this Rotten Tomatoes dataset. Our goal is to label reviews related to movies on a scale of five values (5-class classification problem). Namely, the five classes which we have are: negative, somewhat negative, neutral, somewhat positive and positive.

However, as we mentioned before this task is very challenging.

More specifically,the dataset is comprised of tab-separated files with phrases from the Rotten Tomatoes dataset [8]. Initially, the training data and testing data has been preserved for purposes of benchmarking. However, the sentences have been shuffled from their original order. Each Sentence has been parsed into many phrases through the Stanford parser. Subsequently each phrase has a Phrase Id. Similarly, each sentence has a Sentence Id in order to be able to track which phrases belong to a single sentence. Finally, phrases that are repeated (such as short/common words) have been included only once in our data.

Nevertheless, the most important features of our dataset in which we are going to build on, are the "Phrase" that contains 156060 phrases of user reviews on movies and the "Sentiment" which contains the sentiment of each of the phrases and belongs to one of the five aforementioned labels (0-4). In order for our code to be more efficient, we decided to work with 50000 of the 156060 phrases of the user reviews.

In order to be able to implement our sentiment-analysis approach on this Rotten Tomatoes dataset, two separate approaches have been used. Namely, the first one is a Machine Learning approach, whereas the second one is a Sentiment-Lexicons approach. We implemented both these two ways as this allows us to conduct a thorough analysis of the problem and to make some useful conclusions. A pipeline of our work follows in the next section. Finally, a partition of the five classes can be depicted in figure 1.

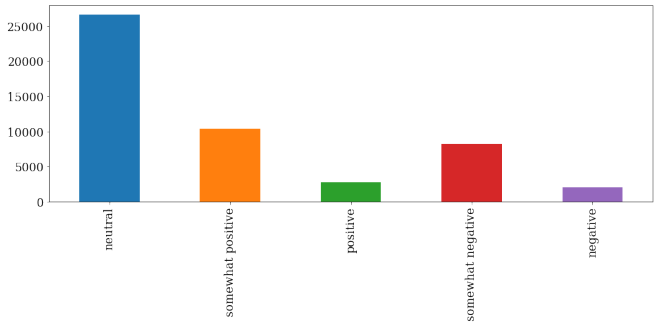

Figure 1: Partition of classes in the dataset

B) Pipeline

A pipeline of our work can be depicted in figure 2.

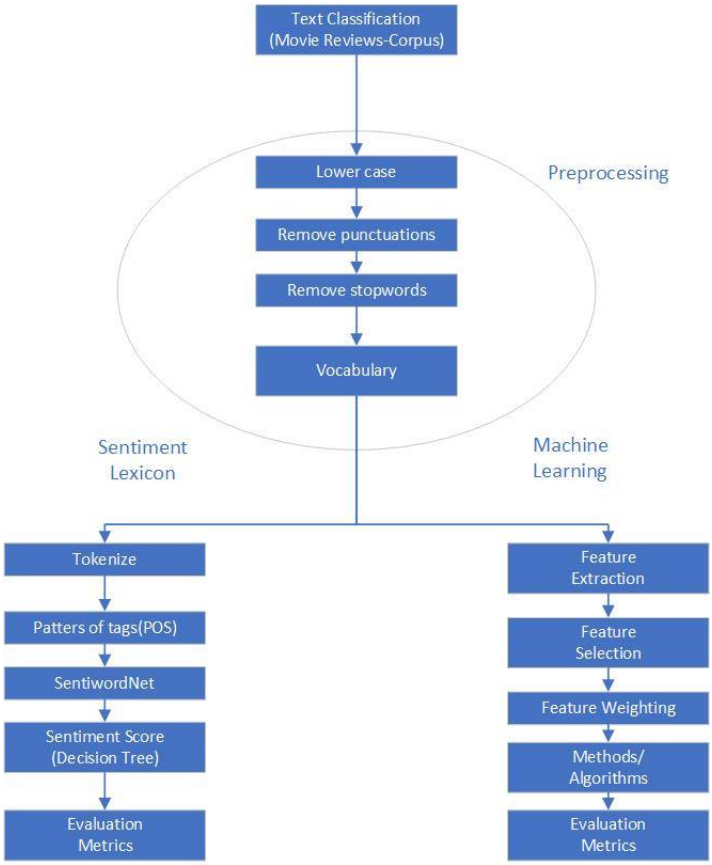

Figure 2: Pipeline of our Work

C) Text Classification-Movie Reviews

As we mentioned before, in this work we are dealing with text classification (textual data) and more specifically we focus on sentiment analysis in movie reviews. The domain of movie reviews is experimentally convenient for two main reasons. Firstly, there is a plethora of collections of such reviews and secondly the sentient of reviews is summarized in a machine-extractable way (i.e number of stars that a movie has). The type of classification, which we are dealing with is a Single-Label Multi-Class (SLMC) classification problem, as we have five classes and each item of the dataset belongs to exactly one class. Also, as a second categorization of our problem we could say that we a have a Hard Classification (HC) in our case, since each item of our dataset belongs to exactly one class as we mentioned and we are not in case of Soft Classification (SC) where we have scores for each class. Furthermore, we are dealing with a Supervised Classification since our training data are all labeled. Finally, as far as the dimensions of the classification are being considered, in our problem, this text classification is performed by sentiment.

D) Preprocessing

After observing the data which we have we went through some stages of preparing and cleaning our data in order to

feed them to our two approaches (Machine Learning and Sentiment lexicons) as they do not all of them provide any meaningful information and to achieve better results. It should be pointed out here, that in a more general case, the pipeline should also include the conversion of the raw review (i.e one paragraph) into sentences. Nevertheless, as we stressed it before, our dataset was from the beginning in a sentence form. There are some basic transformations that are required in order to process text data. Preprocessing is an important step for preparing the data to avoid errors in the classification procedure. First of all, all words are lower cased. After that, all the punctuations are deleted. In the sequel, we also delete all the stopwords (i.e."the", "a", "to"). Finally, all the left phrases are collected and consist our vocabulary, in which we are going to work with.

E) Machine Learning Approach

1) Feature Extraction

Initially, in the Machine Learning approach, after the preprocessing which we performed in our data, we should convert our data into vectors in a common vector space in order to be able to feed them as an input to our learning algorithms, namely to the classifiers which we use. The dimensions of this vector space are going to be our features. Hence, in order to be able to generate this vector-based representation for our vocabulary, feature extraction is performed.

More specifically, in our work we have features which initially represent phrases. Based on the distinction we want to capture (sentiment analysis) we decided to deal not only with unigrams (bag-of-words) but also with n-grams, namely bigrams and trigrams as in the classification by sentiment task, bag-of-words is not enough, and deeper linguistic processing is necessary. This is something which we perceived it in our experiments, as we noticed that our classifiers did not have a good performance by just using bag-of-words. Also, we notice that adding n-grams with $n > 4$ not only it had a higher computational cost, but also is did not contribute in an increase in the performance of the classifiers. Furthermore, we made some experiments in which we used a "summation" of unigrams+bigrams, unigrams+bigrams+trigrams, bigrams+trigrams instead of using separately, unigrams, bigrams and trigrams. The results, of these approaches are going to be discussed in the Experiments section.

Finally, we should stress here that part of speech tagging approach (POS tagging) also have been experimented in the machine learning approach besides in the sentiment lexicons approach. However, after conducting our experiments, no meaningful difference was noticed during the results. Hence, we decided not to include this approach into the rest of our work in the Machine Learning approach and to only use it in the Sentiment Lexicon approach.

2) Feature Selection

Feature selection is an important and useful method in order to improve the accuracy and decrease the computational time and the memory usage of an algorithm. As we use n-grams then this has as result that the length of our vectors (features) to be extremely large. If this is the case, then problems like overfitting and high computational cost arise. In order to be able to prevent them, a feature selection approach is examined. The goal of feature selection approach is to identify the most discriminative features, which have a high contribution in the performance of our classifiers while the less discriminative ones may be discarded. Thus we select a specific subset of terms from our training set and we only use this subset for our classification task. Hence, in this approach we try to measure the discriminative power of each feature and to finally keep only the top-scoring features. The measure which we adopt in order to perform this is the usage of a Chi-Square test.

Furthermore, it would provide a clearer insight to give a brief description of the use of Chi-Square test in feature selection. Chi-Square test examines whether the occurrence of a specific term and the occurrence of a specific class are independent. In order to be able to perform this, each term is evaluated and a score is calculated for each term. A high score indicates that the null hypothesis of independence must be rejected and indeed the occurrence of a term and the class are dependent on each other. Finally, is this is the case, this specific feature is going to be used for the task of classification. Hence, in this manner, we find the most discriminative features for our case.

Finally, we could say that the main advantages of this method are the reduction in the size of the dataset, since we consider only the most discriminative features and, therefore, such an approach not only allows a faster workout, but also it can contribute in improving the accuracy of our classifiers through removing features that can be seen as noise.

3) Feature Weighting

In addition, a Feature Weighting approach is examined. Feature Weighting has to do with the attribution of a value to a feature in a document. In our work, this value is a numeric value which represents the importance of each feature in the sentence which is obtained through a feature weighting technique, namely the TF-IDF technique (Term Frequency-Inverse Document Frequency) which is a text mining technique that used to categorize documents. The main purpose of this method is that tries to emphasize features that occur frequently in a given sentence and simultaneously to deemphasize features that occur frequently in many sentences.

In order to compute this approach in a more efficient way we implemented some intermediate steps. To be more specific, most frequent words from the sentences have been removed as they do not provide meaningful information and thus can bias the result (i.e the word movie). In addition really rare words (very small frequency) have also be discarded as they add some kind of noise. Finally, like in the previous approach (Feature selection) n-grams combinations have also be considered as we stated before are able to catch the overall sentiment of a phrase in a more subtle way in comparison to unigrams.

4) Methods/Algorithms

Initially, we split our dataset into a training and a testing set. A percentage of 80% of our data is used as the training set and the remaining 20% as the test set. Of course, we keep separately the test set from the training set in order to have a fair evaluation of our classifiers. Also, this split is performed

several times in a randomized way in order to be able to cross validate our results. As we mentioned before, in this work we try to implement our sentiment-analysis approach on the Rotten Tomatoes dataset. Six classification algorithms have been examined:Naive Bayes, SVMs, Logistic Regression, Neural Networks, 3-NN and Decision trees. Although, the philosophies behind these six algorithms are quite different, each of them has been shown to be effective in previous similar studies.

Naive Bayes (Generative Machine Learning technique) works on the principle of probabilities and on the Bayes rule. It is very fast in learning and testing as it just counts words, it has low storage requirements, it is more robust to irrelevant features in comparison to the other learning methods which we adopt and we use it as it is a good dependable baseline for text classification even though it does not have the best performance.

SVM algorithm (Discriminative Machine Learning technique) tries to find a margin between the data points for a given dataset. The thickness of this margin is determined by the distance between the support vectors which are data points that are closest to the separating function and lie on the borders of this margin. Through this margin and the support vectors we are able to classify examples. The SVM classifier gives better results because it guarantees optimality. Due to the nature of Convex Optimization, the solution is guaranteed to be the global minimum not a local minimum. Also, it is useful for both Linearly Separable(hard margin) and Nonlinearly Separable(soft margin) data.However, given a large dataset, the train time of SVM could be too slow.

After that we also examined the Logistic Regression classifier (Discriminative Machine Learning technique), which extracts some set of weighted features from the input, uses logarithms and combines them into a linear way as we dealt with a classifiaction problem in sentence level.

Furthermore, in Nearest Neighbor classification, an object is classified by a majority vote of its neighbors and is assigned to the most common class among its k nearest neighbors. However, given a large dataset, the test time of 3-NN could be too slow.

In addition, Decision Trees classifier also was experimented which only considers a subset of features at each node and despite the fact that for large enough amount of data we found that it was a little slow, in overall in had a very good performance.

Finally, we also examined a more complex classifier, neural network and more specifically perceptron (Discriminative Machine Learning technique), and we found that given a large dataset and a very large number of features it had the best performance. However the computational cost in such a case was high.

As an overview and justification for these particular choices of classifiers, we should mention that we took into account the nature of each of these classification schemes, the large amount of data and features which we have in our disposal. Also the fact that we wanted to have a thorough view of the limitations and advantages of these classification schemes in practice. Finally, matters like trade-offs between performance, computational time and memory storage limitations have also been considered.

5) Evaluation metrics

In order to be able to measure the performance of our methods different evaluation metrics have been used. These metrics indicate how efficient a specific method is in order to predict a label for a particular phrase [9]. Our evaluation system integrates four different measures. Namely the accuracy, precision, recall and f-measure have been adopted [10]. More specifically, Accuracy represents the number of correct labeled phrases divided by the total number of phrases that we had to label. Precision, represents the percentage of phrases which have been correctly labeled into a class over those that have been classified in that category. Recall, represents the percentage of all relevant phrases correctly labeled into a class over all relevant words from that class.The range of Precision and recall ranges from 0 to 1. In addition, precision represents a quantitative measure of the system while recall represents a qualitative measure of this system. Finally, F-measure represents the harmonic mean between Precision and Recall. Specifically, this measure penalizes better comparatively with other measures (as arithmetic mean) a system that influences more a metric than the other.

Thus, we could say here that in overall precision means the minimization off false positive results whereas recall concerns the minimization of false negative results. However, neither of these alone is a very good indicator of success. Thus, the metric F-score is a statistical measure that combines the two aforementioned measures into a single one which gives an overall score of accuracy. Hence, for this reason we decided to use all these four evaluation metrics in order to be able to have a thorough view of the performance of the classifiers which we used.

F) Sentiment Lexicons Approach

Also a lexicon based approach was used to classify the results, namely Sentiwordnet. Sentiwordnet is a sentiment lexicon associating sentiment information to each wordnet synset.

Sentiwordnet receives as input a single word and returns a list of synsets. It labels each synset with a value for each category between 0.0 and 1.0. The sum of the three values is always 1.0, so each synset can have a nonzero value for each sentiment because some synsets can be positive, negative or objective depending on the context in which they are used. However in the newest version the objectivity score is omitted. The strategy we followed is this one. Firstly, the already preprocess phrases were tokenized, and then each word was given a tag by applying POS-tagging. After that, each word was given as input to SentiwordNet where we built upon a decision tree to classify each phrase. We considered a score for each synset s to be the positive value minus the negative value, $score(s) = pos(s) - neg(s)$. Then we considered the score of a word w to be the average score of all its synsets s, $score(w) = \frac{\sum_s score(s)}{len(synsets)}$. Finally we decided the final score of a sentence $sent$ to be to the average score of its words w, $score(sent) = \frac{\sum_w score(w)}{len(sent)}$. The result of our algorithm is a score for each sentence. Since that score is normalized it

would be between $[-1, 1]$. In the next step we divided that space in 5 subsequent spaces and each one corresponds to an actual sentiment.

- $[-1, -0.6] \approx$ *negative*
- $[-0.6, -0.2] \approx$ *somewhat negative*
- $[-0.2, +0.2] \approx$ *neutral*
- $[+0.2, +0.6] \approx$ *somewhat positive*
- $[+0.6, +1] \approx$ *positive*

However, after preprocessing it can be noticed that some sentences ended up being empty. We decided the sentiment of an empty sentence to be neutral. The final step was to evaluate the results by means of recall, precision, accuracy, F1 measure.

EXPERIMENTS

Machine Learning Approach

In order to illustrate the use of the above Six classification algorithms: Naive Bayes, SVMs, Logistic Regression, Neural Networks, 3-NN and Decision trees four evaluation metrics have been used. It can be depicted from the results below, that there is suggestive evidence that sentiment categorization (sentiment in movies reviews) is a difficult topic. In this subsection the results and the evaluation metrics about the machine learning approach are presented. We are interested in evaluating the influence of three main factors in each classifier: the Feature Extraction, the Feature Extraction with Feature Selection and the Feature Extraction with Feature Weighting.

Table 1: Average accuracy, precision, recall and f-measure for Bayes Classifier, in percent.

Bayes Classifier	Accuracy	F1	Recall	Precision
Unigrams	60.96	**44.24**	51.04	41.22
Bigrams	56.21	31.83	48.87	30.08
Unig + Big	60.95	43.83	50.92	40.80
Unigrams + FS	**61.46**	42.99	53.44	39.46
Bigrams + FS	56.15	29.37	52.62	28.39
Unig + Big + FS	61.21	41.58	52.73	38.18
Unigrams + FW	55.96	22.66	52.96	24.57
Bigrams + FW	52.99	14.11	30.98	20.10
Unig + Big + FW	56.07	22.85	53.24	24.69

Table 2: Average accuracy, precision, recall and f-measure for Logistic Regression Classifier, in percent.

Logistic Regression Classifier	Accuracy	F1	Recall	Precision
Unigrams	62.93	**45.30**	56.65	41.28
Bigrams	57.50	32.74	55.95	30.55
Unig + Big	**63.02**	45.20	57.04	41.13
Unigrams + FS	62.72	44.48	56.24	40.52
Bigrams + FS	57.67	32.55	56.98	30.43
Unig + Big + FS	62.72	44.49	57.45	40.36
Unigrams + FW	56.74	29.19	51.22	28.39
Bigrams + FW	52.96	14.48	25.62	20.22
Unig + Big + FW	56.73	29.33	51.31	28.46

Table 3: Average accuracy, precision, recall and f-measure for Neural Network Classifier, in percent.

Neural Network Classifier	Accuracy	F1	Recall	Precision
Unigrams	55.97	23.58	20.58	28.29
Bigrams	57.57	29.58	45.87	28.87
Unig + Big	55.47	23.39	20.42	28.06
Unigrams + FS	52.92	13.84	10.58	20.00
Bigrams + FS	56.76	29.09	46.06	28.53
Unig + Big + FS	**60.97**	**39.83**	45.35	38.49
Unigrams + FW	56.45	27.52	54.47	27.81
Bigrams + FW	52.94	14.50	25.33	20.22
Unig + Big + FW	52.92	13.84	10.58	20.00

Table 1 shows that the accuracy in the Bayes Classifier is around 61%. We easily observed that with use of unigrams and feature selection better results are presented in our metrics. In comparison, the results we obtained by comparing the F1-score distributions show that the unigrams are around 44%. The most significant result for the accuracy is observed on Logistic Regression (table 2) classifier for unigrams and bigrams. Moreover, we need to point out that the recall and the precision are not so good in feature weighting. We can see in table 3 that Neural Network Classifier achieves a very good accuracy and F1 score for unigrams and bigrams with feature selection approach to identify the most discriminative features.

Table 4: Average accuracy, precision, recall and f-measure for 3-NN Classifier, in percent.

3-NN Classifier	Accuracy	F1	Recall	Precision
Unigrams	52.25	42.11	49.53	39.84
Bigrams	56.72	34.51	48.76	32.21
Unig + Big	**61.47**	**45.27**	54.20	41.85
Unigrams + FS	61.31	45.00	54.02	41.66
Bigrams + FS	57.16	34.50	50.13	32.18
Unig + Big + FS	61.04	44.60	53.90	41.27
Unigrams + FW	57.04	37.01	25.33	20.22
Bigrams + FW	52.96	14.76	25.02	20.32
Unig + Big + FW	57.13	36.28	47.01	33.79

Table 5: Average accuracy, precision, recall and f-measure for SVM Classifier, in percent.

SVM Classifier	Accuracy	F1	Recall	Precision
Unigrams	**63.62**	**48.83**	55.23	45.58
Bigrams	57.36	35.12	50.88	32.50
Unig + Big	63.62	48.41	55.81	44.88
Unigrams + FS	63.42	47.66	55.88	44.00
Bigrams + FS	57.81	35.04	52.03	32.46
Unig + Big + FS	63.42	47.37	56.54	43.48
Unigrams + FW	56.75	30.58	46.68	29.34
Bigrams + FW	52.96	14.55	25.83	20.25
Unig + Big + FW	56.80	30.85	46.88	29.53

Table 6: Average accuracy, precision, recall and f-measure for Decision tree, in percent.

Decision tree	Accuracy	F1	Recall	Precision
Unigrams	62.76	**50.17**	52.55	48.71
Bigrams	57.45	37.01	50.56	34.25
Unig + Big	62.52	49.48	52.09	47.92
Unigrams + FS	63.32	49.71	53.11	47.85
Bigrams + FS	58.06	36.81	52.61	33.991
Unig + Big + FS	**63.42**	49.72	53.34	47.76
Unigrams + FW	58.89	39.58	51.45	36.40
Bigrams + FW	52.97	14.76	34.19	20.35
Unig + Big + FW	58.89	39.41	51.36	36.27

In table 4 can be observed that the the accuracy and the F1-score for unigrams give the better results for 3-NN Classifier. The most significant result in machine learning approach has been obtained by SVM classifier. From Table 5, three conclusions could be drawn. First, the accuracy of SVM is higher than the others classifiers. Second, SVM classifier achieves a very good accuracy and a F1-score. The last conclusion is that the even in this classifier the Feature Weighting did not yield good results. Finally, in table 6 the average accuracy, precision, recall and f-measure for Decision Tree are presented. Using this table, similar conclusions about Feature Selection and 2-grams could be drawn. In this method the computational cost was very high.

Sentiment-Lexicons Approach

In this subsection the results and evaluation metrics are presented. Table 7 shows the different combinations of tags which we used. The tags we experimented with are the following ones.

Table 7: POS patterns

POS-Tag	**meaning**
JJ	adjective
NN	noun
R	adverb
V	verb

Next the results of our Metrics are presented for each such a combination of POS patterns.

Table 8: Metrics

	JJ+NN	R+JJ	JJ	R+V
Recall	31.60	32.65	33.78	29.49
Precision	22.91	25.54	26.33	21.59
F1	21.06	25.26	26.12	17.90
Accuracy	52.28	52.98	53.228	53.44

As we mentioned before, accuracy represents the number of correctly labeled phrases divided by the total number of phrases that we had to label. Precision, represents the percentage of phrases which have been correctly labeled into a class over those that have been classified into that specific category. Recall, represents the percentage of all relevant phrases correctly labeled into a class over all relevant phrases from that class. F-measure represents the harmonic mean between Precision and Recall.

We can see that our system achieves a very good accuracy and a good recall. The precision and the F1 measure are not so good. It can also be observed that the best combination was verbs and adverbs in terms of accuracy and adjectives in terms of recall, precision and F1 measure.

Finally, in the next tables the Confusion matrices for each such a combination of POS patterns are presented.

Table 9: Confusion Matrix for JJ+NN

	neg	some neg	neutral	some pos	pos
neg	52	260	1677	44	3
some neg	98	670	7130	283	33
neutral	118	963	24475	934	91
some pos	33	247	8967	1047	129
pos	6	36	2321	335	48

Table 10: Confusion matrix for R+JJ

	neg	some neg	neutral	some pos	pos
neg	116	370	1380	148	322
some neg	214	866	6404	632	98
neutral	222	1062	23666	1415	216
some pos	67	417	7864	1664	411
pos	2	67	1820	679	178

Table 11: Confusion matrix for JJ

	neg	some neg	neutral	some pos	pos
neg	153	386	1350	111	36
some neg	244	955	6419	465	131
neutral	273	1054	23936	1049	269
some pos	111	425	8035	1293	559
pos	15	85	1837	532	277

Table 12: Confusion matrix for R+V

	neg	some neg	neutral	some pos	pos
neg	3	72	1849	111	1
some neg	34	250	7558	368	4
neutral	33	237	25622	677	12
some pos	10	108	9436	836	33
pos	0	21	2387	328	10

CONCLUSIONS

Based on the research and the implementation which we conducted, we could say that analyzing and extracting sentiments for text classification is a tricky and not at all easy task. Initially, the current state-of-the-art algorithms for the problem of sentiment analysis were examined in order to discover which are the most efficient approaches. After that, we analyzed our dataset provided by a Kaggle challenge and we extracted some useful statistics (partition of the five classes) as it allows us to visualize and better understand our data. In the sequel, in our project we tried to investigate two totally different approaches, namely one Machine Learning and one Sentiment Lexicons approach. We performed techniques like preprocessing, feature extraction, feature selection, feature weighting and POS tagging in order to be able to improve our results. After that in the first case, a variety of classification schemes such as Naive Bayes, SVMs, Logistic Regression, Neural Networks, 3-NN and Decision trees were adopted and in the second case a Sentiwordnet approach was used. In the sequel, four metrics, namely Accuracy, Precision, Recall and F1-measure were used in order to be able to evaluate our results. Finally, the results and the interpretation of our results was followed. As an overall comment, we could state that we were able to achieve very good results with respect to the difficulty of this multi-class problem and indeed to classify the majority of the reviews to the correct class.

As we mentioned above, a challenging aspect of this problem that distinguishes it from a classic topic-based classification is that sentiment can be expressed in a subtle manner. Hence, in this kind of tasks we have to face problems like sarcasm and ambiguity which are so difficult even for humans to be able to detect them. Thus, as a future work, we could say that if we want to further improve our results, we should be able to devise an approach that deals with these kinds of constraints. Also, as an avenue for future work we could mention an implementation of our approach not only on a review level but also on an aspect level. We think that such an approach is going to be a very interesting and demanding challenge.

Finally, a Deep Learning approach which takes into account Convolutional Neural Networks (CCNs) and Recurrent Neural Networks (RNNs) instead of the Machine Learning approach which we implemented, could possible give better results and it would be a worth mentioning approach of a future work project.

REFERENCES

1. Shirani-Mehr, Houshmand.*"Applications of Deep Learning to Sentiment Analysis of Movie Reviews."* Available at http://cs224d.stanford.edu/reports.html [July 2015]

2. Singh, V.K., Piryani R., Uddin A., Waila, P, *"Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification,"* published in Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on 22-23 March 2013, pp. 712-717

3. A. Hamouda and A. Rohaim, *âĂIJReviews classification using sentiwordnet lexicon,âĂİ* in Journal on Computer Science and Information Technology (OJCSIT), Vol. (2), No.(1), 2011

4. Vishwanathan, S. (2010), *âĂIJSentiment Analysis for Movie Reviews,âĂİ* Proceedings of 3rd IRF International Conference

5. Pang, B. and L. Lee: 2005, *âĂŸSeeing stars: Exploiting class relationships for sentiment categorization with respect to rating scalesâĂŹ*. In: Proceedings of the 43rd Meeting of the Association for Computational Linguistics (ACL 2005). Ann Arbor, US, pp. 115âĂŞ124

6. Kim, Jungi, Jin-Ji Li, and Jong-Hyeok Lee. *Discovering the discriminative views: Measuring term weights for sentiment analysis*. in Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP (ACL-2009). 2009.

7. L. Zhuang, F. Jing, X.-Y. Zhu, and L. Zhang, *âĂIJMovie review mining and summarization,âĂİ* in Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM), 2006.

8. https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data

9. D. Manning, P. Raghavan, H. SchÃijtze, *Introduction to Information Retrieval*, Cambridge University Press, 2008, ISBN-10: 0521865719

10. D. Manning, H. Schutze, *Foundations of Statistical Natural Language Processing*, MIT Press, ISBN: 987-0-262-133360-9, 1999;

PS:A link to a repository that contains the software we created and the dataset we used is available at:
`https://github.com/ktouloumis/NLP`