Pre-processing Framework for Twitter Sentiment Classification

Elias Dritsas[1], Gerasimos Vonitsanos[1], Ioannis E. Livieris[2], Andreas Kanavos[1,2], Aristidis Ilias[1], Christos Makris[1] and Athanasios Tsakalidis[1]

1. Computer Engineering and Informatics Department
University of Patras, Patras, Greece
eldritsas@gmail.com, {mvonitsanos, kanavos, aristeid, makri, tsak}@ceid.upatras.gr
2. Computer and Informatics Engineering Department
Technological Educational Institute of Western Greece, Antirrion, Greece
livieris@teiwest.gr

Abstract. Twitter Sentiment Classification is undergoing great appeal from the research community; also, user posts and opinions are producing very interesting conclusions and information. In the context of this paper, a pre-processing tool was developed in Python language. This tool processes text and natural language data intending to remove wrong values and noise. The main reason for developing such a tool is to achieve sentiment analysis in an optimum and efficient way. The most remarkable characteristic is considered the use of emojis and emoticons in the sentiment analysis field. Moreover, supervised machine learning techniques were utilized for the analysis of users' posts. Through our experiments, the performance of the involved classifiers, namely Naive Bayes and SVM, under specific parameters such as the size of the training data, the employed methods for feature selection (unigrams, bigrams and trigrams) are evaluated. Finally, the performance was assessed based on independent datasets through the application of k-fold cross validation.

Keywords: Classification, Microblogging, Pre-processing, Sentiment Analysis, Supervised Machine Learning, Twitter

1 Introduction

The rapid development of modern computing systems along with Internet access and high communication capabilities, has turned these schemes into an integral part of human everyday life. Nowadays, users can express their personal opinion on any matter whenever they wish as well as share their thoughts and feelings. It is no coincidence that most websites encourage their users to review their services or products while social media accounts have significantly increased.

In particular, a user through websites can be informed, express their personal views on a variety of topics and simultaneously interact with other users. Hence, this kind of interaction produces a large amount of data that is of particular interest for further process and analysis. Companies have started to poll these

microblogs to get a sense and understand the general sentiment towards their product. Often, these companies study user replies and in following reply on the corresponding microblogs. The challenge is to build tools that can detect and summarize an overall sentiment, so that valuable conclusions and information on various kinds of issues can be drawn. For example, one can consider demographic and social conclusions, information of economic nature such as the prevailing view of a product or service, or even results of political content.

Sentiment analysis constitutes a subtle field in information mining. It is considered a computational analysis and categorization of opinions, feelings and attitudes that are drawn up in text format. The use of Natural Language Processing techniques is sought through polarity performance, while polarity can be characterized by many different classes. Specifically, the positive and negative terms, which correspond to the positive or negative view that a user has, are utilized in terms of a specific event or topic.

It is widely noted that the emotional analysis has many applications, as an individual may have a view on a huge range of issues of a different nature, such as economic, political, religious, and so on. For this reason, the positive and negative classes are not the only ones used, as aforementioned. It constitutes a basic way of studying and analyzing such data. Indeed, in recent years, the great volume of raw information generated by Internet users has increased the interest in processing such data. In other words, it is the process of mining and categorizing subjective data so as to extract information about the polarity and overall emotion expressed in natural language in text form [**?**].

Data mining is considered an important part of the data analysis [**?**]. It largely consists of collection, processing and modeling of data and is aimed at the objectives shown in Figure **??**. Its characteristics are the export of information from a dataset, its following transformation in a comprehensible structure with the use of various techniques (machine learning, statistics, etc.) which facilitate analysis, and finally the conclusions export as well as the decision-making.

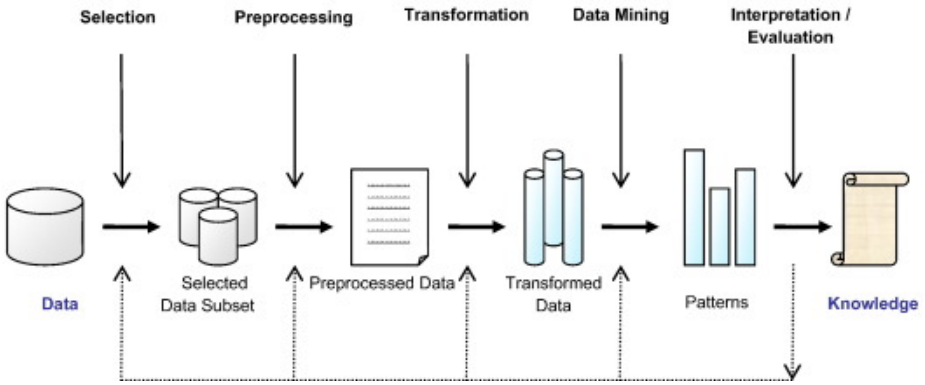

Fig. 1. KDD process for knowledge mining from data [**?**]

In this present work, the main contributions concern the following aspects. Concerning data pre-processing techniques, tweets face several new challenges due to the typical short length and irregular structure of such content. Hence, data pre-processing constitutes a crucial step in the corresponding field of sentiment analysis, since selecting the appropriate pre-processing methods, the correctly classified instances can be increased [**?**]. To properly implement the data analysis process, it is necessary to process the raw data collected in a variety of ways. Initially, it should be decided what data will be used depending on the purpose of the analysis. In following, it is necessary to eliminate the abnormalities and deal with incorrect values and/or incomplete inputs. Subsequently, the processed data are modified in a form suitable for mining them. Therefore, we focus on designing an efficient pre-processing tool which facilitates the sentiment analysis conducted based on supervised machine learning algorithms. Another contribution is the application of Latent Dirichlet Allocation (LDA) based probabilistic system to discover latent topics from the conversations in connection to the event we take into account.

The proposed research is organized as follows: Section **??** describes the background topics and the challenges faced, while Section **??** explains the process of information retrieval from Twitter platform. The same section analyzes the tool for data pre-processing, which is the main contribution of this work. Furthermore, Section **??** presents the experimental results and main conclusions extracted from the study and the analysis. Finally, Section **??** portrays conclusions and draws directions for future work.

2 Related Work

Sentiment Analysis can be considered as a field of data mining, which intends to determine the polarity of a subjective text, through the use of various algorithms. In the recent years, this particular branch of science has started to gain increasing interest both from academic and organizations perspective; users started to study the emotions around a subject for a variety of reasons. This interest is boosted by the rapid growth of users participating in social media, which is a modern world phenomenon. Some researches that have played a key role in the evolution and importance of emotional analysis are presented below.

Pang and Lee [**?**] introduce techniques and challenges in the application of emotional analysis, while at the same time a plethora of datasets are widely used. Also, the authors show how both the frequency of the terms and the n-gram feature selection affect the result. In addition, within the framework of the system developed, the incidence of the individual condition and in following the posts splitting into n-grams was taken into account.

Of particular interest is the application of emotional analysis to different hierarchical levels. Pang et al [**?**] study the effectiveness of document-level for a large number of critically acclaimed films from the popular website IMDB. Turney et al [**?**] examine the polarity of a file through its proposals. Phrases that include adjectives and/or adverbs, or other features and parts of speech

that are highly likely to express the author's emotion, are selected. Concretely, user reviews on a variety of topics such as movies, travel destinations, banks and cars, were utilized. Wilson et al [**?**] analyzed the views of the MPQA corpus along with a set of data containing journal articles from different sources that have been rated in terms of their emotion. An important part of their work was the separation of neutral phrases in polar and objective and then, the polarity analysis of subjective phrases to extract the overall feeling of the text.

Furthermore, the emotional analysis can also be applied in the field of economic interest, in sets of journalistic content and critical films [**?**,**?**,**?**], as well as for political aspects [**?**,**?**]. Initially, authors in [**?**] studied reviews for mobile applications and in following exported important features for their nature. Then, in [**?**], product reviews to identify consumer's sentiment in terms of certain characteristics and the products themselves were analyzed. Authors in [**?**] constructed a model for emotional analysis based on travelers reviews about the destinations they visited. In a similar way, a system that analyzes the sentiment of publications of real-time Twitter users to predict the results of the 2012 presidential election in the United States of America was created [**?**]. Finally, in [**?**], researchers used messages (not real-time data) that included reference to German political parties in 2009.

Previous works regarding emotional content are the ones presented in [**?**,**?**,**?**], in which the authors presented various approaches for the automatic analysis of tweets and the recognition of the emotional content of each tweet based on Ekman emotion model; based on Ekman emotion model, the existence of one or more out of the six basic human emotions (Anger, Disgust, Fear, Joy, Sadness and Surprise) is specified. Finally, a novel distributed framework implemented in Hadoop as well as in Spark for exploiting the hashtags and emoticons inside a tweet is introduced in [**?**]. Moreover, Bloom filters are also utilized to increase the performance of the proposed algorithm.

3 Tools and Environment

The pre-processing methods evaluated by the current research are three different data representations, namely unigrams, bigrams and trigrams. Two well-known machine learning algorithms were selected for classification, namely Naive Bayes and Support Vector Machine (SVM) as shown in the following section.

3.1 Twitter

The platform that is being studied in this work is Twitter. This is a platform for posting posts, exchanging messages between users, and modifying their private profiles according to their needs. There is the possibility of communicating links, images and audiovisual material to the posts. Twitter has gained considerable interest on a global scale, due to the services it provides its users with.

A special feature that makes emotional analysis quite difficult in its context is the small length of suspension that it allows its users. It is therefore perceived

that studying the polarity of users' publications, beyond the general challenges and difficulties faced, is even more difficult due to their limited length.

3.2 Publications Mining Tools

The mining of posts was done using the Tweepy library which through the Twitter interface allows managing a user's profile, collecting data by optionally using certain search keywords, and finally creating and studying a stream of posts over a specific time interval.

In this work, posts were stored in a CSV file, where rows contain the posts that were extracted, while the columns contain the values of the different attributes of each post (e.g., date, text, username, etc.).

Useful tools in the context of this work were the following:

1. The Natural Language Tool Kit is a natural language processing library, which offers classification, parsing, tagging, and clipping stemming possibilities.
2. Scikit-Learn is a library that addresses the implementation and development of machine learning techniques and text editing tools. This library interacts with the Python, NumPy and SciPy computational and scientific libraries enhancing its efficiency and speed significantly.
3. Pickle is a library that converts objects in a form understandable only by the Python language, in order to limit the space they occupy in memory; this is due to certain features (e.g. JSON format) to be stored and reused whenever necessary. In this work, this library was used to allow the usage of objects returned after the classifier training, without retraining, whenever necessary.

3.3 Pre-processing Scheme

In order to facilitate the mining process of the collected data, it is necessary to apply several pre-processing steps [**?**]. The main parts of this process are the following .

Pandas Library: It was utilized to facilitate the management of input files having the components of the publications. Then, a dataframe was created only with the most important components for our analysis, i.e. records were removed, either with incorrect values or the ones not rated.

Regular expressions: They were utilized to remove the urls and references to other users' username. Also, it was possible to find and replace/remove alphanumeric characters that match a predefined search pattern and remove unnecessary spaces. The repetition of suffix characters that were used for emphasis reason and numeric characters, which do not facilitate Sentiment Analysis, were removed as well.

Emoticons: Emoticons are characters, such as punctuation and parentheses, which in turn form representations of expressions of the human face, e.g. cheerful person {:-)}, but also different representations that play an important

role in analyzing the feelings of each publication. Emoticons are widely used in social media, especially on the Twitter, to express feelings and impressions in a short way. Therefore, a set of regular expressions containing a large part of these representations was created. Additionally, a set of widely used unofficial abbreviations was generated in order to replace words that users make up. For example, the lol expression, which was replaced by its equivalent full form, namely, laugh out loud.

Autocorrect library: This library uses a list of words found in recognized dictionaries, and given an input word, compares its similarity to the words on that list. If the input word is correctly spelled, it is returned as a whole. If it is not correctly spelled, then its similarity is checked with the words in the list; if its similarity is greater than a certain threshold, then it is replaced by the word in the list. Otherwise, it is returned as a whole, without any changes.

Pycontractions library: It detects a set of successive characters in which the apostrophe is contained and replaces it with the full form of expression. Expressions with the {'s} special character complicate emotional analysis since they express two or more words, which makes tokenization, as well as, normalization, particularly difficult. For effective mining, contractions are written in their original form no matter how complicated it is. In case the set has a single possible replacement, then the expression is transformed into its original full form. Using a grammatical controller and Word Mover's Distance [**?**], a method of calculating the distance between the original text and the texts produced, called "compatibility" metric, is derived. The substitution applied is the one with the highest value.

Emoji library: They are Unicode characters in the form of an icon representing face expressions, and many kinds of objects. This part differentiates our work from others, as in most other studies, emoticons and emojis are not taken into account. However, they are used in most publications in opinion mining. This library uses a mapping list, as created by the Unicode Consortium. Unicode characters included in a text are reviewed, and if they are in this list, they are replaced with the text form of their representation.

Part-of-Speech Tagging: Each word of the text is tagged according to the part of the speech that it constitutes (adverb, verb, object). This process uses the context of the text to be analyzed as well as a set of aggregated elements (corpus), to evaluate and attribute the part of speech to the particular term being studied.

Lemmatization: It is the process in which lexical and morphological analyses of words are taken into account in order to remove complex suffixies and to retrieve the lexical form of the term. It is applied after POS tagging and facilitates the emotional analysis through the application of machine learning algorithms. In the context of this work, POS tagging labels are Penn Treebank format.

Tokenization: It is the separation of sentences into a list of symbol terms that can be used to reform the original sentence. Both emoticons and emojis that have now been converted to text characters are taken as tokens, without

being divided into individual characters or punctuation marks. The tokenization process is applied to all sentences, and their terms are stored in the same token list. Essentially, the tokens list is created for each post. Once the publication's details are now in a list in the order they appear in it, some more conversions are made to optimize and improve the viewing process.

Punctuation: These are also tokens of the list. Generally, punctuation marks do not attach any emotional significance to the publication and thus, they are removed.

Stopwords: These are words appearing very often, without expressing some form of feeling. The reason why such words are removed is because the whole attempt is to examine meaningful words in order to determine the overall emotion expressed in publication.

3.4 Features

N-grams are one of the most common structures used in text mining and natural language processing fields. They constitute a set of co-occuring words within a given window. In addition, as already known, a Markov assumption is the assumption that the probability of a word depends only on the previous word. So, Markov models are these probabilistic models that with their use, the probability of a future aspect without looking too far into the past can be predicted. The most popular ones are the bigrams, which search for one word from the past, the trigrams, which search for two words from the past, and the n-grams, which search for $n1$ words from the past.

The "bag-of-words" approach is considered a very simple and flexible representation of text that describes the occurrence of words within a document. It involves a vocabulary of known words as well as a metric of the presence of these known words. The model considers only the existence of the known words in the document, and not the exact place where to be found in the document. The intuition is that documents are similar if they have similar content.

The Summed Term Frequency constitutes the sum of all the term frequencies in the documents. In the proposed paper, it is utilized as

$$SummedTermFrequency = \sum_{d \in D} TF_{n-gram} \tag{1}$$

In addition, the "Apply Features" method has been taken into consideration in order to obtain a feature-value representation of the documents. Concretely, this method is used in order to apply a "positive" or a "negative" label to each feature of the training data.

3.5 Topic Modeling

One other aspect we want to take into consideration in our proposed work is the verification of whether all the posts discuss the specific topic. Topic modeling considers a document as a "bag-of-topics" representation, and its purpose is

to cluster each term in each post into a relevant topic. Variations of different probabilistic topic models [**?**], [**?**] have been proposed and LDA [**?**] is considered to be a well known method.

Concretely, the LDA model extracts the most common topics discussed that are represented by the words most frequently used, by simply taking as input a group of documents. The input is a term document matrix, and the output is composed of two distributions, namely document-topic distribution θ and topic-word distribution ϕ. EM [**?**] and Gibbs Sampling [**?**] algorithms were proposed to derive the distributions of θ and ϕ. In this paper, we use the Gibbs Sampling based LDA. In this approach, one of the most significant steps is updating each topic assignments individually for each term in every documents according to the probabilities calculated using Equation 2.

$$\mathbb{P}(z_i = k|z_{-i}, w, \alpha, \beta) \propto \frac{(n^{-i}_{(k,m,\cdot)} + \alpha)(n^{-i}_{(k,\cdot,w_i)} + \beta)}{n^{-i}_{(k,\cdot,\cdot)} + V\beta} \tag{2}$$

where $z_i = k$ shows that the i_{th} term in a document is assigned to topic k, z_{-i} signifies all the assignments of topic except the i_{th} term, $n^{-i}_{(k,m,\cdot)}$ is the number of times that the document d contains the topic k, $n^{-i}_{(k,\cdot,w_i)}$ is the number of times that term v is assigned to topic k, V represents the size of the vocabulary as well as α and β are hyper-parameters for the document-topic distribution and topic-word distribution respectively.

The number of the Gibbs sampling iterations performed for every terms in the corpus is N; after this component, the document-topic θ and topic-word ϕ distributions are estimated using Equations 3 and 4 respectively.

$$\hat{\theta}_{m,k} = \frac{n_{(k,m,\cdot)} + \alpha}{K\alpha + \sum_{k=1}^{K} n_{(k,m,\cdot)}} \tag{3}$$

$$\hat{\phi}_{k,v} = \frac{n_{(k,\cdot,v)} + \beta}{V\beta + \sum_{v=1}^{V} n_{(k,\cdot,v)}} \tag{4}$$

4 Evaluation

In Table **??**, the two datasets studied as well as their characteristics are presented. There are 5 different categories whereas the first dataset contains tweets for all of them and the second dataset contains tweets for the 3 of them. The total number of tweets studied per each dataset is also considered.

The first dataset consists of tweets about self-driving cars[1]. The sentiment is categorized into 5 categories, ranging from very positive to very negative. The second dataset consists of the feelings that travelers have in February 2015 towards the problems of each major U.S. airline[2]. The sentiment in the tweets of this dataset is categorized as positive, neutral or negative for six US airlines.

[1] `https://www.kaggle.com/c/twitter-sentiment-analysis-self-driving-cars`
[2] `https://www.kaggle.com/crowdflower/twitter-airline-sentiment`

Table 1. Datasets Details

Sentiment	**Selfdriving Cars Dataset**	**Airlines Dataset**
Positive	1262	2363
Slightly Positive	1452	
Neutral	4245	3099
Slightly Negative	1498	
Negative	1076	9178
Total Number of Tweets	9533	14640

The results of our work are presented in the following Tables **??** to **??**. The Accuracy, in terms of percentage, is used as the evaluation metric of the two different algorithms (Naive Bayes and SVM) for the different setup (Unigrams, Bigrams and Trigrams). Also, the percentage of training and test set is taken into account when considering the two datasets.

In Table **??**, the results of the RapidMiner platform[3] are presented. We have used RapidMiner as a baseline to emphasize the improvement of our proposed methodology. Furthermore, in RapidMiner we cannot include features that are utilised in our paper, such as emojis and emoticons, etc. We observe that SVM performs better than Naive Bayes for the three different setups and for both datasets. Secondly, in both datasets, Unigrams and Bigrams achieve better accuracy than Trigrams; this is expected as tweets usually have small length due to the number restriction that characters have and thus, Trigrams cannot be considered as a qualitative metric.

Table 2. RapidMiner Results - Accuracy

Setup	**Selfdriving Cars Dataset**	**Airlines Dataset**
Naive Bayes (Unigrams)	62.13	64.95
Naive Bayes (Bigrams)	64.37	62.40
Naive Bayes (Trigrams)	58.89	50.66
SVM (Unigrams)	69.56	75.50
SVM (Bigrams)	71.78	73.20
SVM (Trigrams)	71.49	70.95

In following, Table **??** presents the results for different ratio of training versus test set. We have utilised three different cases, with training set having values equal to 70, 75 and 80 whereas test set has values equal to 30, 25 and 20 respectively. Worth noting is the fact that our proposed methodology outperforms the results from RapidMiner as regarding Selfdriving Cars Dataset, for both classifiers and the three different setups, the accuracy has lower value equal to 72% and higher equal to 80%. On the other hand, for the Airlines dataset, we observe high rate fluctuations as the higher and the lower percentages are de-

[3] `https://rapidminer.com/`

picted. Concretely, Naive Bayes (Trigrams) achieves the lowest accuracy with almost 60% and Naive Bayes (Unigrams) achieves the highest accuracy with 85% (for training-test ratio equal to 80-20). Furthermore, we notice that for all cases, as the percentage of training set increases, so does the accuracy. This is something that we expect as higher values of training set increase the classifier's results.

Table 3. Accuracy for different Training - Test Set ratios

	Selfdriving Cars dataset			**Airlines dataset**		
Setup	70 - 30	75 - 25	80 - 20	70 - 30	75 - 25	80 - 20
Naive Bayes (Unigrams)	71.99	72.73	74.04	82.60	83.72	85.18
Naive Bayes (Bigrams)	77.83	79.49	80.76	77.78	78.55	80.94
Naive Bayes (Trigrams)	76.57	77.93	79.44	59.45	59.70	60.72
SVM (Unigrams)	73.50	74.45	74.15	77.45	77.57	78.44
SVM (Bigrams)	76.43	77.77	78.76	68.90	69.43	69.56
SVM (Trigrams)	75.70	77.27	78.67	64.94	65.28	65.82

Finally, Table **??** presents the Accuracy results when splitting with 10-Fold Cross-Validation. The concept of using this technique is that important information can be removed from the training set. In addition, this method is simple to understand and generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split. As in Table **??**, Naive Bayes outperforms SVM, while it achieves the higher percentage with value equal to 85%. What is more, for both datasets, Naive Bayes has values close to 79% whereas SVM has different values for the three setups.

Table 4. 10-Fold Cross-Validation

Setup	**Selfdriving Cars Dataset**	**Airlines Dataset**
Naive Bayes (Unigrams)	78.66	85.38
Naive Bayes (Bigrams)	79.92	79.99
Naive Bayes (Trigrams)	79.29	74.26
SVM (Unigrams)	73.04	77.26
SVM (Bigrams)	76.59	69.32
SVM (Trigrams)	75.93	65.25

5 Conclusions and Future Work

In this paper, we proposed a pre-processing framework for Twitter sentiment classification. We chose Twitter because of tweets' short length and content's diversity. We used supervised machine learning techniques for the analysis of

the raw data in the user posts and incorporated emojis and emoticons if order to enrich our features. Furthermore, we applied the Latent Dirichlet Allocation (LDA) based probabilistic system to discover latent topics from the conversations. Two popular classifiers (Naive Bayes and SVM) were used for three different data representations (unigrams, bigrams and trigrams) in order to perform our experiments in two datasets.

In the near future, we plan to extend and improve our framework by exploring more traits that may be added in the feature vector and will increase the classification performance. Moreover, we plan to compare the classification performance of our solution with other classification methods. Another future consideration is the adoption of other heuristics for handling complex semantic issues, such as irony that is typical of messages on Twitter.

Acknowledgement

Elias Dritsas was funded by General Secretariat for Research and Technology (GSRT) and Hellenic Foundation for Research and Innovation (HFRI) and supported by University of Patras. Andreas Kanavos, Aristidis Ilias and Christos Makris are co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program Research and Innovation Strategies for Smart Specialisation - RIS3 of "Partnership Agreement (PA) 2014-2020".