Twitter Sentiment Analysis

Advanced Internet Computing

Martin Kaufleitner
Vienna University of
Technology
Austria
e1027229@student.tuwien.ac.at

Martin Kaufleitner
Vienna University of
Technology
Austria
e1027229@student.tuwien.ac.at

Martin Kaufleitner
Vienna University of
Technology
Austria
e1027229@student.tuwien.ac.at

Martin Kaufleitner
Vienna University of
Technology
Austria
e1027229@student.tuwien.ac.at

ABSTRACT

Twitter Sentiment Analysis is a crucial task in today's scenarios where opinions gain more weight for further investigations and developments of companies. In this paper we want to provide an overview of the state-of-the-art regarding sentiment analysis of Twitter messages, including the extensive growth and the new beneficial possibilities to classify messages of the Twitter platform. The experimental evaluation of our dataset, its classification results and findings do not contradict to any existing results from other scientific work.

1. INTRODUCTION

Social media has revolutionised the way in which people communicate. Gaining information from social networks is beneficial for analysis of user opinion for example measuring the feedback on a recently released product, looking at opinions concerning upcoming elections or the enjoyment of an ongoing event. Sentiment analysis is a relatively new area, which deals with extracting user opinion. Since Twitter has well established as social media platform for personal continuous news and opinions flow from all over the world, it also became an eye catcher for different organisations to get feedback of their product applying sentiment analysis on tweets. As a consequence this trend influenced various marketing and sales stragies and created a new market for companies that are specialise on sentimental analysing tweets, like âĂę An example of a positive sentiment is, âĂđtraveling is funâĂIJ alternatively, a negative sentiment is âĂđitâĂŹs a horrible day, i am not going outsideâĂIJ. Furthermore, emoticons such as âĂđ:-)âĂIJ is a postive and âĂđ:-(âĂđ is a negative expression which also influence the result of sentiment analyse. Objective texts are deemed not to be expressing any sentiment, such as news headlines, for example âĂIJcompany shelves wind sector plansâĂİ. There are many ways in which data can be leveraged to give a better understanding of user opinion. Sentiment analysis of tweets aims for analysing whether the global opinion located at Twitter concerning an expression is positive or negative. This can be done in different ways, e.g. different grammatic and language rules. Such problems are at the heart of natural language processing (NLP) and data mining research. Generally, âĂę about twitter api..

2. PRE-PROCESSING

Since all different kinds of classifiers try to extract, evaluate and interpret different features, the amount and quality of those features is essential for a good result of the classification. We want to examine some state of the art pre-processing and feature reduction techniques, which presented themselves as very promising and effective. While examining them, we will discuss, which ones of the techniques we have also implemented in our Sentiment Analyzing Software and how we did this.

2.1 Twitter Domain

Since we are dealing with twitter messages, we have to deal with some specialties, which need some extra preparation during the pre-processing and feature reduction steps. This specialities are stated in almost all papers, which deal with twitter sentiment check, so we want to have a look at them. We only consider the most significant specialities, a more detailed description can be found e.g. et al [1]

- Limited Length
  Twitter messages have a limited length of 140 characters. This fact leads to the occurrence of many abbreviations and slang formulations. Some examples are "OMG" for "Oh my god" or "FTW" for "For the win".
- Emoticons
  Also the occurrence of emoticons needs special treatment. There are different approaches, how to take them into account which we want to discuss.
- Casual language
  Since there are no guidelines or rules how to formulate

tweets, they contain very casual language. An example would be a tweet like âĂIJOMG IâĂŹm soooo huuuuungry!!!!âĂİ. The multiple occurrence of letters in the words âĂIJsoooooâĂİ and âĂIJhuuuuuungryâĂİ would blow up the amount of features, since âĂIJhungryâĂİ, âĂIJhuungryâĂİ, âĂIJhuuungryâĂİ and so on would all be treated as different words. Also misspelled words lead to the same problem. We will discuss some ways how to address this problem.

- Links
  Many tweets contain links to other web pages or articles. Those links can mislead the interpretation of a tweet.
- Usernames, Re-Tweets and Hashtags
  Symbols like RT for Retweet, @ for linking other users or a hashtag often occur in tweets and therefore ask for special treatment.

2.2 Methods

Now let us have a look at some different methods according to the pre-processing step which deal with the problems occurring in the stated twitter domain.

2.2.1 *Usernames, Hashtags, Links*

Most approaches like [5] remove all usernames, hashtags and links and replace their occurrences with according tags like <username>. This very simple method reduces the amount of features enormously, since otherwise, each username, which does not give any information about a sentiment, would be treated as a feature. Furthermore, all tweets, which are re-tweets are deleted, since otherwise the original tweet would be over weighted and maybe falsify the result. These methods are stated in almost all twitter sentiment analyzing approaches and since they are very simple, we implemented all of them. But despite of the described processing in [2] we did not tag tweets when we e.g. remove a link. Unfortunately the paper does not describe in detail, how they use this tag information afterwards, so we decided to leave it out.

2.2.2 *Emoticons*

Regarding emoticons, there are different approaches which could be followed. The first one doesnâĂŹt treat them in any special way, so they are considered like any other word and are listed in the features e.g. [3]. The first basic step is to classify them into e.g. positive and negative ones and then replace them. For example the emoticons :) :D and XD are all replaced with <positiveEmoticon> and :(:âĂŹ(and so on are replaced with <negativeEmoticon>. This helps reducing features, but if you let emoticons in your data, they may get over fitted, that means, that the occurrence of an emoticon will almost exclusively decide on the classification of the tweet. This is especially the case, when you retrieved your training data by searching for tweets with emoticons as noisy labels. Therefore there are approaches, like [5], which delete the emoticons completely from the training data and only allow them during classification or some approaches which even remove them from the test data. Since we obtained our test set classification by querying twitter for tweets containing :) for positive and :(for negative, we had to delete them from our training set to prevent over fitting. For the test data, we replaced them with the according positive and negative tags in order to reduce features and then treated them as a normal feature, since removing them would result in a loss of important sentiment information.

2.2.3 *Casual language and misspelled words*

The problem with casual language and misspelled words is, that the number of features grows very vast. To get rid of misspelled words, spell checking algorithms are available, which try to find the intended meaning of a word and replace them. According to words like ”huuuungry” there are again two different approaches. Either all multiply occurrences are simplified to at most two letters (”huungry”) and then treated as an own feature [1], since multiple letters may express the sentiment stronger, or they are then spell-checked and replaced by the corresponding word like ”hungry” in this case. Since we think, that the additional information, given by using multiple letters should not be lost, we implemented the first approach and therefore the @TODO: Keine ahnung wie wir das gemach haben âĂŞ Also putting all letters in lower case helps reducing unnecessary features.

2.2.4 *Stopwords*

So calledstop words like ”and”, ”the”, ”for”, ”a” and so on do not provide any information about the sentiment of a tweet and are therefore unnecessary features which can be removed as proposed in almost all papers. Databases containing this words can be found in the internet.

2.2.5 *Abbreviations*

Abbreviations could either be replaced by their original words, or used as their own feature. Most paper describe the replacement and therefore we also followed this strategy in our project. We used a simple Abbreviations database, tuned it, and then replaced all abbreviations according to this list.

2.2.6 *Tokenization*

After preparing the data, the text has to be separated into different features. We decided to split the sentences at white spaces and punctuations using a Twitter Tokenization library. We did not consider special words like ”DonâĂŹt” or ”IâĂŹll” to stay together, as proposed et al [4].

3. CLASSIFICATION

3.1 Overview

In the context of sentiment analysis, classification is intended to determine the sentiment of a piece of text (such as a Twitter status), e.g. how positive or negative the sentiment behind the content is. This is also referred to as ”sentiment polarity classifications”, to differentiate it from other sorts of sentiment. A variety of other metrics exists, such as political affiliation (e.g. conservative/liberal) or various kinds of approval/disapproval [5]. Sentiment classification is a classical natural language processing (NLP) problem, and its high complexity is largely founded in the complexity and ambiguity of human speech itself (cite???).

The already high complexity of sentiment analysis is compounded by the typical terseness of tweets (thus containing few relevant words), as well as frequent use of informal language and (sometimes ambiguous) abbreviations [4]. Sentiment analysis at the phrase level is more complicated than at the document level, and has been developed only more recently. Furthermore, spelling errors and informal spelling are frequent. [1].

Furthermore, tweets often combine different or even diametrically opposing sentiments in close proximity. While these are easily separated for humans, this task is very complex for machines [7].

On the other hand, microblog content might also offer some clues which are unlikely to appear in a more formal medium. For example, emoticons are frequently used to aid classification [1, 4], or to tag an unclassified training set [2].

In the literature, many approaches to classifying tweets using various combinations of features and machine learning algorithms are described.

Generally speaking, sentiment analysis poses two separate challenges, which are both critical for the quality of the results. The first is to recognize and extract features which are suitable indicators of sentiment. The second part consists of passing them – as part of either training or evaluation – to some mechanism for machine learning in order to assign a single sentiment value to a piece of text.

3.1.1 Features

A variety of features can be extracted from messages and used for classification purposes. Unigrams, bigrams and larger n-grams represent words or other sorts of tokens (e.g. emoticons and exclamation marks). Sentiments can be assigned to individual words using special dictionaries and used as features [4].

In addition to simple lists, processing features as part of more complex structures is also possible. In [1], a tree kernel is used to perform classification over many different features at once. These features include integers, reals and booleans, and form a hierarchical structure (the tree). A Support Vector Machine is used for learning, with the results outperforming a second implementation using unigrams. Even better results were obtained by combining both.

A fundamental problem of all sentiment analysis methods is that the meaning of words often depends strongly on context [7], therefore simple features may be strongly misleading. The simplest example is negation using a preceding ”not”, which completely inverts the sentiment of the word it is connected to.

Approaches to solving this problem include the use of bigrams, unigrams with explicit negation features [2] or Part-of-Speech (POS) tagging [4]. However, the success of such methods seems to vary significantly with the precise application area and implementation, and such more advanced methods may actually yield worse results than simple unigrams. POS tagging in particular appears to consistently decrease accuracy [1, 4, 2].

Frequently, different features are combined (e.g. unigrams + bigrams [2]), leading to better overall results.

Sarcasm is frequently used on platforms such as Twitter, but resolving or even detecting sarcasm is very difficult. In [3], an attempt was made to detect sarcasm in Tweets, but results were not very satisfying. However, humans did not perform much better at this tasks, leading the authors to conclude that extensive knowledge about the context and participants of a conversation would be required for both humans and algorithms.

Care must also be taken to exclude features which may bias the classification of a message. When searching for tweets containing a certain term, for example, it might be beneficial to exclude this term from the features [2]. Otherwise, the overall sentiment for this token might influence the sentiment result of the current tweet, where exactly this sentiment in *relation to the term* is desired.

Semantic analysis can be used to solve many of the aforementioned problems [6]. References to entities are classified into ”semantic concepts”, allowing for conclusions over broader classes of entities. The authors achieved an improvement of 6.5% over unigrams and 4.8% over POS features.

3.1.2 Machine Learning

For machine learning, usually some sort of supervised learning is used, where a pre-labelled training set is used for learning (and optionally a second set for evaluation). Once training is complete, new data can be classified. Popular machine learning techniques used in this field include Naive Bayes and Maximum Entropy Classifiers, as well as Support Vector Machines [7].

In [2], three algorithms (Naive Bayes, Maximum Entropy and SVM) are compared. Training was done on emoticon data, resulting in accuracy of over 80% for all three approaches.

Kouloumpis et al. [4] use AdaBoost as a learning algorithm in their implementation. They found a combination of n-grams, pre-tagged words and microblogging-specific features to yield the best results.

Coming up with a suitable dataset to train the machine learning algorithm can be quite challenging on its own. One possibility is the use of manually tagged datasets, but these are usually quite limited in size due to the human effort required to build them. Alternatively, the training set can be automatically tagged using some presumably reliable indicator, such as emoticons [2].

3.2 Our Approach

In our work, we used a strongly simplified approach which focuses on individual tokens (unigrams). All aforementioned issues such as negation or sarcasm notwithstanding, the sentiment of a tweet is often determined with reasonable accuracy by a small set of keywords signalling strong emotion.

Semantic or grammatical analysis was deemed to be for too complex, and simplified methods were judged to not provide notable improvements and lead to false positives. We also did not differentiate between objective and subjective valuation, since both are likely relevant for the given use case.

TODO/FIXME: Three different machine learning algorithms were tested: Naive Bayes, IBk and SMO. Training of the support vector machine was significantly slower than for the other algorithms, while results were ??????TODO.

3.3 Conclusions

A frequent observation is that more involved classification methods lead to only marginally better or even worse results (see e.g. [1]).

TODO: extend

4. CONCLUSIONS

5. REFERENCES

[1] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. Sentiment analysis of twitter data. In

Proceedings of the Workshop on Languages in Social Media, pages 30–38. Association for Computational Linguistics, 2011.

[2] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[3] R. González-Ibáñez, S. Muresan, and N. Wacholder. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 581–586. Association for Computational Linguistics, 2011.

[4] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538–541, 2011.

[5] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[6] H. Saif, Y. He, and H. Alani. Semantic sentiment analysis of twitter. In *The Semantic Web–ISWC 2012*, pages 508–524. Springer, 2012.

[7] G. Vinodhini and R. Chandrasekaran. Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6), 2012.