Multilanguage sentiment-analysis of Twitter data on the example of Swiss politicians

Lucas Brönnimann
University of Applied Science Northwestern Switzerland, CH-5210 Windisch, Switzerland
Email: lucas.broennimann@students.fhnw.ch

Abstract
Sentiment analysis is an important tool in the study of social media and is very well researched for texts written in English. However, in many cases multi-language text analysis is required and a simple translation of the text to English would result in inferior solutions. A novel field of application is the analysis of the communication in social media by politicians in a country with multiple national languages, such as Switzerland.
A machine learning approach using large amounts of tweets written by Swiss politicians is applied to determine the affiliation with a party for anyone writing about political subjects on Twitter. While text similarity alone achieves acceptable results, it can be shown that the combination with a multi-language sentiment analysis for the key topics improves the accuracy of such an approach.
The paper also describes the developed sentiment algorithm which employs emoticons as a universally comprehensible clue on whether a given text is positive or negative. This allows for a language specific acquisition of a sentiment lexicon which can be used with a simple algorithm to determine the sentiment of Messages on Twitter in their respective language.

Sentiment Analysis, Multilanguage, Big Data, Twitter, Politics, Switzerland, Machine Learning

Introduction
Twitter is a social network site where users post messages up to 140 characters which are called tweets and can be read by anyone. According to Twitter there are over 200 million active users and more than 400million tweets are written daily as of March 2013 [1].
Various researchers have analyzed the huge amount of data that is available on Twitter. The focus is often set on analyzing the content of the tweets, in some cases combined with additional metadata such as time or the geolocation of the tweets, if available. Other researchers focus on analyzing the sentiment of tweets and therefore mine for opinions or use this data to predict stock market fluctuations [2] [3] [4].
Sentiment analysis or opinion mining tries to find emotion hidden behind the text. Complex algorithms aim to determine whether the author of a tweet was sad, happy, angry or something else. It is of special interest to determine whether the author writes positive or negative about whatever the subject of the tweet is, as this can be used for marketing purposes, managing reputations or analyzing popularity [5] [6].
At the core of most of these algorithms is a lexicon with features, consisting of a varying amount of words, rated with a sentiment between positive and negative or something similar, such as the probability for the word occurring in a tweet with a given emotion [7]. These lexica are typically created by using a large corpus of documents from a single domain, or by extending already known lists of words with other English synonyms and close words [8]. A lot of research has been done with English tweets, especially through analyzing the hash tags of Twitter [9], but it has to be taken into account that only approximately 38% of the tweets are in English [10]. While there are English tweets to be found in every country, the vast majority will be in the national language(s).
Since most processes are based on English words, they cannot be used to analyze texts in other languages. To generate a sentiment lexicon with words of any language, a language independent way to detect emotion or to differentiate between positive and negative tweets is required.

Emoticons as language independent indicators for sentiment
Of all the text messages sent on Twitter each day, approximately 5 - 10% contains an emoticon. These are used to express a wide range of different emotions and determine in most cases correctly the sentiment the author conveyed with his message, making it ideal to use as noisy labels for distant supervised learning [11] [12]. Of particular interest are emoticons that express positive or negative emotions, such as sadness, anger, happiness or delight. Some examples of typical western emoticons can be found in Table 1. It is to be kept in mind, that eastern languages more often use non-rotated symbols, which would also have to be included if these languages are of interest, but can mostly be ignored otherwise.

Negative	:(:-(:{ :-|| :@ :'(:/ D:< ಠ_ಠ
Positive	:-) :) :o) :-D :D 8-D XD =)

Table 1 Typical emoticons to express positive and negative sentiment in western languages

The Twitter streaming API allows any developer to access part of the huge amount of data on Twitter and the search can additionally be specified as to only contain tweets with certain keywords or emoticons and as of May 2013 even the language can be specified. If enough time is available, building a database with a huge amount of tweets can be done very easily and the requirements to the processor are such that even a simple single-board computer such as a Raspberry Pi is capable of fetching millions of tweets every day.
While positive emoticons are more common, it is still advisable to generate a corpus of approximately the same size for positive and negative texts to simplify further analysis. At this stage, it is also important to have the language information of every text available or separate every language that is to be analyzed into its

own text corpus, which can be done through any Language Detection Algorithm with reasonable accuracy [13]. In this paper, a corpus of at least 1 million tweets per analyzed language is used, up to over 40 million tweets for English, gathered between February and June 2013.

Building the sentiment lexicon

After the creation of the corpus with positive and negative documents, the goal is to find features determining whether a tweet is positive or negative. While there are various approaches, it has been shown that even a simple unigram analysis of the words provide high accuracy [11] [14]. Therefore the first step should be to tokenize the corpus and count the frequency of each word. Optionally the words can also be stemmed at this step, but this requires a stemmer for each language that is to be analyzed. If enough data is available a stemmer might even be adverse to a high accuracy.

The result of this process is a list for each desired language with words and their occurrence in positive and in negative texts which can directly be used as a word-sentiment association lexicon. Table 2 shows a small extract of such a document in English based on approximately 20 million positive and 20 million negative texts. While for most common words the amount of negative and positive texts is about the same, there are certain exceptions, e.g. tweets that contain the word "you" have a bigger chance of being positive, while tweets containing the word "my" are more often negative. Various sentiment algorithms ignore these words as stop words [15], but this table clearly shows that some information can be found in them. However, words such as "happy" typically appear about ten times more often in positive texts than in negative text and are therefore still much more important for a sentiment algorithm. Table 2 also contains the words with the highest and lowest ratio between positive and negative occurrences.

Word	Negative	Positive
rt	3991393	5245004
to	2972964	2742714
you	2500653	3358939
my	2380459	1852518
unusable	1871	23
heartbreaking	6619	158
preordering	17	4659
Prosperity	66	1183

Table 2 Extract of the sentiment lexicon with the most common words

Besides information about the sentiment, this list also gives a very good overview of the use of language on twitter. Not surprisingly it shows that the most common word is "rt", which is used to mark a tweet as a copy of a post from another user, a so called retweet. Abbreviations and various misspellings are also quite common on Twitter, which means limiting the list to words appearing in a dictionary of the given language is not recommended. Instead, pruning should consist of only removing uncommon words that appeared once or twice in the collected tweets for performance reasons, as well as filtering out unusable words such as "8ilZsnP7rD" which are parts of a link to an image or a website.

In the next step, known language-specific information can be added to each sentiment lexicon. There are some manually created lists of positive and negative words available that should be considered for this, especially the list by Hu and Liu [16] which is not domain-specific and can therefore be used for various applications. A partially automated translation of this list into the target language can also prove useful. Other additions can be language-specific lists of insults or swear words to further indicate a negative sentiment. When combining this information with the sentiment lexicon, all words which appear in the gathered lists as positive should have their value for positive occurrences increased, while for negative words the value for negative occurrences should increase. The exact values should correlate to the size of the sentiment lexicon in the given language.

Given an equal amount of positive and negative training data, there is an additional issue to be kept in mind, namely the way people use the emoticons. In most languages, positive emoticons are about four times more often than negative ones. People are frequently using positive emoticons to convey that they are happy even if the text itself would not indicate it in any way. This means that a classification of a word into the negative class is stronger than one into the positive class where it is more "washed-out". Either the algorithm used for classification of a text will reflect that later on, or some adjustments have to be made directly in the sentiment lexicon.

The strength of this effect can be measured by using a test set with an identical amount of negative and positive tweets in the given language. Ideally, a classifier based on the given sentiment lexicon should have approximately the same amount of false positives for negative tweets, as well for positive tweets in this situation. As an example, in case of the naïve Bayes algorithm [17] with an English test data set, about two thirds of the tweets were classified as negative and one third as positive. The most direct and simplest way to adjust for this problem proved to be multiplying the amount for "total amount of positive texts" by a value of 1.3, resulting in an approximately even distribution between tweets classified as positive and negative. This value varies between languages, so if possible, the test should be repeated with every used language.

There is one further adjustment to the sentiment lexicon that might be useful. Since rarely used words are not getting deleted, they have a high chance to become victims of random effects with a strong impact on the classification. For example any uncommon word which was supposed to be neutral, but through random effects had a high amount of co-occurrences with positive smileys is not a reliable indicator of sentiment. To counter this effect, a small number, proportional to the total amount of tweets in the given language should be added to all negative and positive occurrences.

Usage

The existence of a word-sentiment association lexicon alone is not enough, but it is one of the most important factors in sentiment analysis as even simple algorithms can provide a very high accuracy with a good model [18]. To determine the sentiment of a new tweet, the proposed process would be as following:

1) Detect the language. If no model exists for the recognized language, try to translate the tweet to a known language or discard
2) Tokenize the text with the same procedure as in the training data
3) Use the naïve Bayes algorithm and assign a value between 0 and 1, a lower score being more negative and a higher one more positive
4) Classify the tweet as positive, negative or neutral by using a predefined range for neutral values.

In previous work about sentiment analysis, various classifiers have been analyzed. Typically, a support vector machine provides the best results [19], but other approaches are not far

behind. Training a support vector machine with millions of tweets and using it on a large data set is a very time intensive task, which is a big drawback. Since it has been shown that the training data set is much more important than the actual algorithm [14] it makes sense to look for alternatives.

For this project, the naïve Bayes algorithm proved to be a very viable classification algorithm. One of its advantages is the ease of use, as it has the ability to directly utilize the sentiment lexicon simply with the values "amount of times the feature appeared in positive texts" and "amount of times the feature appeared in negative texts", meaning no expensive conversation is necessary. The result of this process is a value between 0 and 1, which is much more useful information than simply a classification into "positive" or "negative". It can be used to determine the strength of the sentiment behind a tweet and also to recognize neutral tweets without them actually being present in the training data set. Because of this extremely useful ability and a high overall performance, a standard naïve Bayes classifier will be used in this paper, applying the probabilities for each word to be in the positive /negative training data, as gathered in the sentiment lexicon. This paper will from now on simply refer to it as the classification algorithm.

Before using the classification algorithm, the range for neutral values has to be defined. It allows shifting the focus between precision and recall, as the more tweets are classified as neutral, the higher the precision for the classification of the other tweets will be. If the size of the range would be set to zero, no text could ever be classified as neutral. This would be fatal, as most of the messages on Twitter do not contain any form of recognizable sentiment and only state simple facts or consist of a query without detectable emotion.

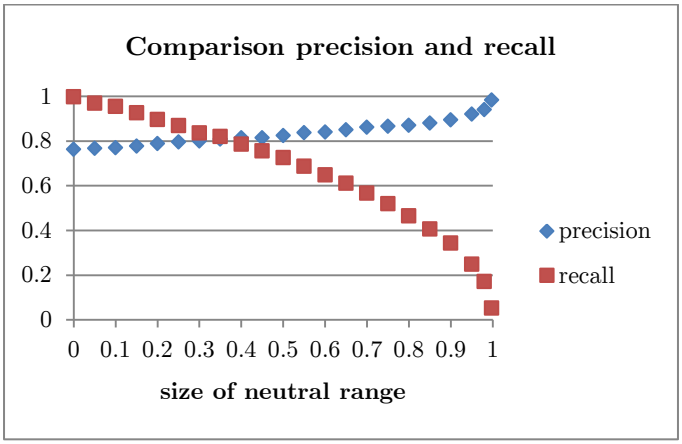

Figure 1 As more texts are classified as neutral, the prediction for the other tweets will improve

A simple benchmark application uses a single tweet as input and gives the naïve Bayes classification to positive, negative or neutral as output. For this purpose a test data set has been manually compiled that consists of approximately 1200 English, 800 French, 200 German and 200 Italian tweets. Figure 1 shows the behavior of the values for precision and recall for different ranges in the English test data set and can be used to help find an appropriate range for the neutral value. While it may be hard to perfectly optimize the value, 0.4 has proven to be a good approximation for most tweets. It means, a range from 0.0 to 0.3 is classified as negative, 0.3 to 0.7 as neutral and 0.7 to 1.0 as positive. Using these suggested values, a classifier can reach up to 81% accuracy in English, 80% in French, 75% in German and 72% in Italian.

The difference in accuracy can partially be explained by the more than 20 times bigger training data set available in English compared to the ones in other languages. In addition, most of the existing sentiment lexica are in English and lose value when translated to other languages. There are also some differences in the quality of the test data set, mostly based on the fact that humans don't always agree on the sentiment of a given tweet. For a further analysis of the accuracy it would be advisable to use a bigger group of people and only add tweets to the data set if they unanimously agree on the sentiment.

The algorithm and the benchmark can now be used to proof that there is a relevant difference in accuracy between classifying a text in the language of the classifier and an automatically translated text. Figure 2 shows the result of the benchmark above, including data sets with translated tweets. As can be seen, the benchmark continuously performs the best with the original language, even though the classifier otherwise performs significantly better with English or French tweets.

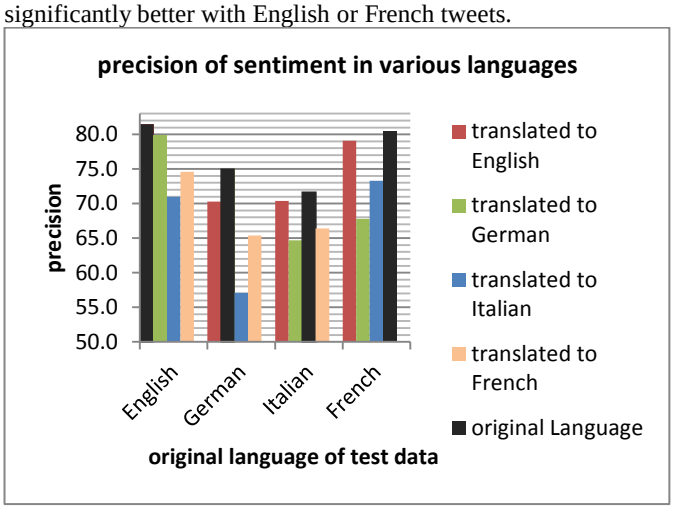

Figure 2 Test data sets with translated tweets perform significantly worse than data sets in the original language

Analysis of Swiss politicians

Switzerland is in the unique position that the language used on Twitter consists of an assorted distribution between German, French, Italian and English with a small amount of tweets in Swiss German or other languages.

As a possible application for sentiment analysis, a classifier for political parties in Switzerland is analyzed. The Swiss parliament has 79 politicians with a twitter account and most of them use it actively to convey some of their political agendas or to communicate directly with the populace. To analyze the use of twitter by these communications, a set of over 25'000 tweets written by members the Swiss parliament has been collected between January 2012 and September 2013. There are exactly 50 politicians with at least 20 tweets during that time frame (one tweet per month on average) belonging to one of the 5 major political parties. Table 3 gives an overview of these parties.

Party	Member	On Twitter	Tweets
SP	56	22	12277
CVP	31	11	4854
FDP	30	7	1683
SVP	46	6	2647
Grüne	15	4	4687

Table 3 Break down of the politicians in the Swiss parliament

It is assumed that there are similarities between tweets of members of the same party. These similarities can be used to train a model based on the same followers and discussion

partners. Other similarities may be visible in the frequent topics members of a given party use in conversation. These approaches are very typical and achieve acceptable results. However, the work on the Multilanguage sentiment analysis leads to a particularly interesting research question: *Can sentiment analysis be used to assign twitter profiles to political parties?*

It stands to reason that a party which supports a given initiative will talk positive about it, while another party opposing the initiative might talk mostly negative about it. We analyze this question using the Multilanguage sentiment analysis described in the first part of this paper.

Test setup

The setup consists of 50 politicians out of 5 parties: SP, CVP, FDP, SVP and Grüne. To determine the accuracy, a leave-one-out cross-validation (LOOCV) test will be used, meaning the model will be trained with the tweets of 49 politicians and the classifier will determine the party of the last one. The test will be repeated 50 times until each politician has been classified exactly once, equivalent to a 50-fold cross validation. Accuracy is defined as the percentage of Twitter accounts correctly assigned to the correct party.

To confirm or deny the research question, the accuracy of four different procedures will be compared against each other. A random classifier, taking into account the ratio of politicians between the parties in the test set, serves as a first baseline. The second approach is to analyze the frequent topics of the politicians through TF-IDF. Because of poor performance with texts of varying language, the texts will be translated to German if necessary.

For the third approach the sentiment of each tweet will be calculated in the original language. The algorithm then calculates the average sentiment for the 100 most often used words, while ignoring all stop words. This results in a vector similar to the TF-IDF, but containing values for the sentiment instead.

Finally the last two approaches will be combined to create a new vector with a value for the sentiment and the TF-IDF value for each keyword. The goal in this case is to have a well-known algorithm that is already performing very well and see if a combination with sentiment can improve it, therefore confirming the hypothesis.

In all three latter cases the difference between the vector of the tested politician and each party will be used to determine the similarity with the party, resulting in 5 values between 0 and 1 for each politician. The higher the value, the higher the probability he belongs to this specific party. For each tested Twitter account the result of any classifier will be a probability matrix, containing the before-mentioned values as calculated by the respective algorithm, normalized to add up to 100%. The account will be correctly sorted if the probability for the actual party is the highest.

To gain additional information from a test with a given algorithm, it makes sense to analyze the calculated probabilities for all 5 parties. The ones for the parties the politician is not a member of should be as low as possible compared to the probability for the correct party. For this purpose, an arbitrary error-value will be calculated in the form of $\sum \left(\frac{p_i}{p_{correct}}\right)^3 - 1$. This makes sure, a strong classification into the correct party will have a very low error value, while a situation where the classifier assumed the correct party was the least likely results in a very high error value. To give a sense of scale, politicians that were classified correctly have an error-value of 3.5 or lower. Higher values indicate a wrong classification.

Results

The results in Table 4 show that sentiment Analysis can successfully be used to improve the quality of the classifier. Unfortunately, given the small test sample of only 50 politicians, it is not significant on the 5% threshold, but the results are still promising for further research.

The combined approach achieves an accuracy of 54%, compared to the 48% of the TF-IDF approach. While the absolute value might not sound like very much, it has to be kept in mind that there are 5 potential parties a politician in the test set can belong to. Additionally in 70% of the cases the algorithm will put the correct party at least in second place and in 78% at least in third. Only in very rare cases the actual party membership will be calculated as the least likely.

Algorithm	Accuracy	Avg. error
Random	36.9%	$3.4*10^{10}$
TF-IDF	48%	4.347
Sentiment of Key topics	44%	3.915
Combined	54%	3.684

Table 4 Evaluation of the politician classifier shows the importance of combining various classification methods

The average error further strengthens the hypothesis. Interesting to note is the better performance of the sentiment classifier compared to TF-IDF in this benchmark. This means that while the sentiment classifier was worse in predicting the actual membership, it was slightly better in predicting which parties the tested politician would most likely not belong to.

Further analysis of the classification process also shows some interesting facts about the use of Twitter by the members of the parliament. Left leaning parties are typically much more active users of social media than right leaning parties. However, nearly all politicians who actively post messages use Twitter nearly exclusively to talk about political topics.

The error value can also be used to determine how well the classification works on average per party and shows the Twitter users with the biggest differences to their own party. It seems to be easier to correctly determine the political leaning of a left leaning politician while the FDP as a middle-right party seemed to be the hardest to classify. Figure 3 shows the error value for each politician during the cross-validation and the color indicates his membership in a party.

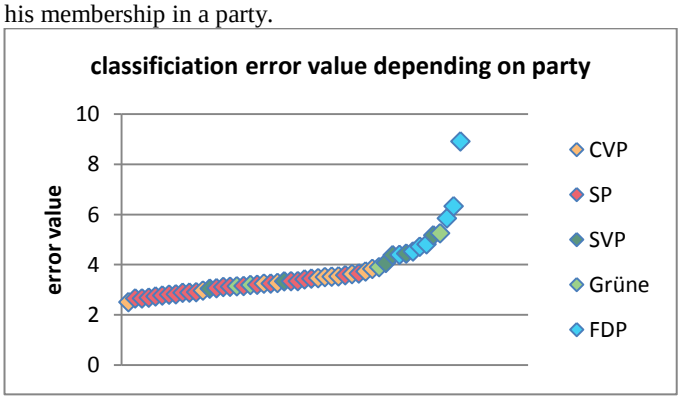

Figure 3 Members of some parties are harder to classify than others.

Conclusion

Sentiment analysis can be used to improve the classification of politicians, even in cases where languages other than English are used. The successful classification shows, that politicians in the same party tend to use twitter mostly similar to other politicians of the same party and will not only talk about similar topics, but also express the same or at least similar opinions.

To create the sentiment lexicon needed for such a task, emoticons proved to be a valuable language independent indicator of sentiment, although the data has to be carefully analyzed, processed and refurbished. A sentiment classifier trained in the language of the test data will outperform a classifier using translated tweets in terms of accuracy and speed.

Research Issues

Sentiment analysis is just one additional feature that can be used to improve a classifier for political parties using Twitter data. Other options could include the follower network of a Twitter user or his most frequent communication partners. It would also be useful to repeat the experiment on politicians in other countries.

As for the sentiment analysis itself, it has not been taken into account, that there might be better sentiment classifiers in the English language which require some additional Natural Language Processing tasks [20]. These tasks however are very dependent on the language and typically only work well in English. Therefore it still stands to reason that in some cases it would be preferable to translate the text to English for better results in the sentiment.

Currently, sentiment Analysis is constantly being researched, but a lot of topics such as sarcasm or figures of speech are poorly understood even by modern day computers and algorithms. Often it is also desired to not only classify the tweets by positive and negative sentiment but instead by a more granular scale with emotions ranging from "sad" to "angry" and more. Typical approaches to these tasks require the documents to be in English and while some may easily be converted to other languages, the results are to be further analyzed.

References

[1] H. Tsukayama, "The Washington Post," 21 March 2013. [Online]. Available: http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter. [Accessed 3 June 2013].

[2] J. Bollen, H. Mao and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science,* pp. 1-8, January 2011.

[3] S. Chung and S. Liu, "Predicting Stock Market Fluctuations from Twitter," Berkeley, California, 2011.

[4] J. Rittermann, M. Osborne and E. Klein, "Using Prediction Markets and Twitter to Predict a Swine Flu Pandemic," *1st International Workshop on Mining Social Media,* 9 October 2009.

[5] H. Saif, Y. He and H. Alani, "Semantic sentiment analysis of twitter," *The 11th International Semantic Web Conference (ISWC 2012),* 11-15 November 2012.

[6] R. Feldman, "Techniques and Applications for Sentiment Analysis," *communications of the acm 56 (4),* pp. 82-89, April 2013.

[7] N. Kaji and M. Kitsuregawa, "Building Lexicon for Sentiment Analysis from Massive Collection of HTML," *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL),* p. 1075–1083, 2007.

[8] G. Qiu, B. Liu, J. Bu and C. Chen, "Expanding Domain Sentiment Lexicon through Double Propagation," *International Joint Conference on Artificial Intelligence (IJCAI-09),* 2009.

[9] S. M. Mohammad, "Emotional Tweets," *First Joint Conference on Lexical and Computational Semantics (*SEM),* p. 246–255, June 2012.

[10] K. Leetaru, S. Wang, G. Cao, A. Padmanabhan and E. Shook, "Mapping the global Twitter heartbeat: The geography of Twitter," *First Monday 18.5,* 2013.

[11] A. Go, R. Bhayani and L. Huang, *Twitter Sentiment Classification using Distant Supervision,* Stanford, 2009.

[12] P. D. Boer, *private communication,* Brugg, 2013.

[13] S. Nakatani, "Language Detection Library," Cybozu Labs, Inc, Tokyo, Japan, 2010.

[14] M. B. Florian Lüscher, "Sentiment Analysis," Fachhochschule Nordwestschweiz, Windisch, Schweiz, 2013.

[15] J. Bollen, H. Mao and A. Pepe, "Modeling Public Mood and Emotion: Twitter Sentiment and Socio-Economic Phenomena," *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media,* 2011.

[16] B. Liu and M. Hu, "Opinion Mining, Sentiment Analysis, and Opinion Spam Detection," 2012. [Online]. Available: http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon. [Accessed 10 January 2013].

[17] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier," *Machine Learning,* no. 29, pp. 103-137, 1997.

[18] B. Pang, L. Lee and S. Vaithyianathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," *Proceedings of the 2002 Conference on Empirical Methods in Natural Language,* 2002.

[19] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai and A. Kappas, "Sentiment Strength Detection in Short Informal Text," *Journal of the American Society for Information Science and Technology 61.12,* pp. 2544-2558, 2010.

[20] B. Liu, "Sentiment Analysis and Subjectivity," *Handbook of natural language processing 2* , p. 568, 2010 .