nlp.cs.aueb.gr: Two Stage Sentiment Analysis

Prodromos Malakasiotis, Rafael Michael Karampatsis, Konstantina Makrynioti and **John Pavlopoulos**
Department of Informatics
Athens University of Economics and Business
Patission 76, GR-104 34 Athens, Greece

Abstract

This paper describes the systems with which we participated in the task Sentiment Analysis in Twitter of SEMEVAL 2013 and specifically the Message Polarity Classification. We used a 2-stage pipeline approach employing a linear SVM classifier at each stage and several features including BOW features, POS based features and lexicon based features. We have also experimented with Naive Bayes classifiers trained with BOW features.

1 Introduction

During the last years, Twitter has become a very popular microblogging service. Millions of users publish messages every day, often expressing their feelings or opinion about a variety of events, topics, products, etc. Analysing this kind of content has drawn the attention of many companies and researchers, as it can lead to useful information for fields, such as personalized marketing or social profiling. The informal language, the spelling mistakes, the slang and special abbreviations that are frequently used in tweets differentiate them from traditional texts, such as articles or reviews, and present new challenges for the task of sentiment analysis.

The Message Polarity Classification is defined as the task of deciding whether a message M conveys a positive, negative or neutral sentiment. For instance M_1 below expresses a positive sentiment, M_2 a negative one, while M_3 has no sentiment at all.

M_1: GREAT GAME GIRLS!! On to districts Monday at Fox!! Thanks to the fans for coming out :)

M_2: Firework just came on my tv and I just broke down and sat and cried, I need help okay

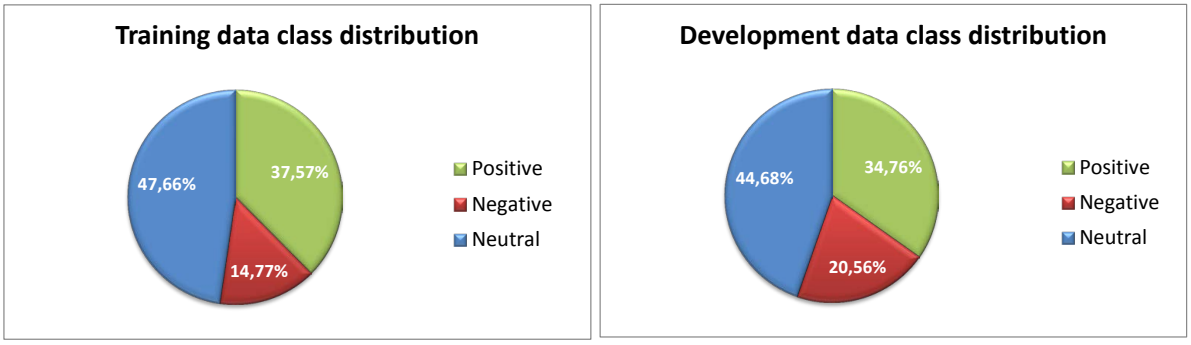
M_3: Going to a bulls game with Aaliyah & hope next Thursday

As sentiment analysis in Twitter is a very recent subject, it is certain that more research and improvements are needed. This paper presents our approach for the subtask of Message Polarity Classification (Wilson et al., 2013) of SEMEVAL 2013. We used a 2-stage pipeline approach employing a linear SVM classifier at each stage and several features including bag of words (BOW) features, part-of-speech (POS) based features and lexicon based features. We have also experimented with Naive Bayes classifiers trained with BOW features.

The rest of the paper is organised as follows. Section 2 provides a short analysis of the data used while section 3 describes our approach. Section 4 describes the experiments we performed and the corresponding results and section 5 concludes and gives hints for future work.

2 Data

Before we proceed with our system description we briefly describe the data released by the organisers. The training set consists of a set of IDs corresponding to tweet messages, along with their annotations. A message can be annotated as positive, negative or neutral. In order to address privacy concerns, rather than releasing the original Tweets, the organisers chose to provide a python script for downloading the data. This resulted to different training sets for the participants since tweets may often become



(a) (b)

Figure 1: Train and Development data class distribution.

unavailable due to a number of reasons. Concerning the development and test sets the organisers downloaded and provided the tweets. [1] A first analysis of the data indicates that they suffer from a class imbalance problem. Specifically the training data we have downloaded contain 8730 tweets (3280 positive, 1289 negative, 4161 neutral), while the development set contains 1654 tweets (575 positive, 340 negative, 739 neutral). Figure 1 illustrates the problem on train and development sets.

3 System Overview

The system we propose is a 2–stage pipeline procedure employing SVM classifiers (Vapnik, 1998) to detect whether each message M expresses positive, negative or no sentiment (figure 2). Specifically, during the first stage we attempt to detect if M expresses a sentiment (positive or negative) or not. If so, M is called "subjective", otherwise it is called "objective" or "neutral".[2] Each subjective message is then classified in a second stage as "positive" or "negative". Such a 2–stage approach has also been suggested in (Pang and Lee, 2004) to improve sentiment classification of reviews by discarding objective sentences, in (Wilson et al., 2005a) for phrase-level sentiment analysis, and in (Barbosa and Feng, 2010) for sentiment analysis on Twitter messages.

[1] A separate test set with SMS messages was also provided by the organisers to measure performance of systems over other types of message data. No training and development data were provided for this set.

[2] Hereafter we will use the terms "objective" and "neutral" interchangeably.

3.1 Data Preprocessing

Before we could proceed with feature engineering, we performed several preprocessing steps. To be more precise, a twitter specific tokeniser and part-of-speech (POS) tagger (Ritter et al., 2011) were used to obtain the tokens and the corresponding POS tags which are necessary for a particular set of features to be described later. In addition to these, six lexicons, originating from Wilson's (2005b) lexicon, were created. This lexicon contains expressions that given a context (i.e., surrounding words) indicate subjectivity. The expression that in most context expresses sentiment is considered to be "strong" subjective, otherwise it is considered weak subjective (i.e., it has specific subjective usages). So, we first split the lexicon in two smaller, one containing strong and one containing weak subjective expressions. Moreover, Wilson also reports the polarity of each expression out of context (prior polarity) which can be positive, negative or neutral. As a consequence, we further split each of the two lexicons into three smaller according to the prior polarity of the expression, resulting to the following six lexicons:

S_+ : Contains strong subjective expressions with positive prior polarity.

S_- : Contains strong subjective expressions with negative prior polarity.

S_0 : Contains strong subjective expressions with neutral prior polarity.

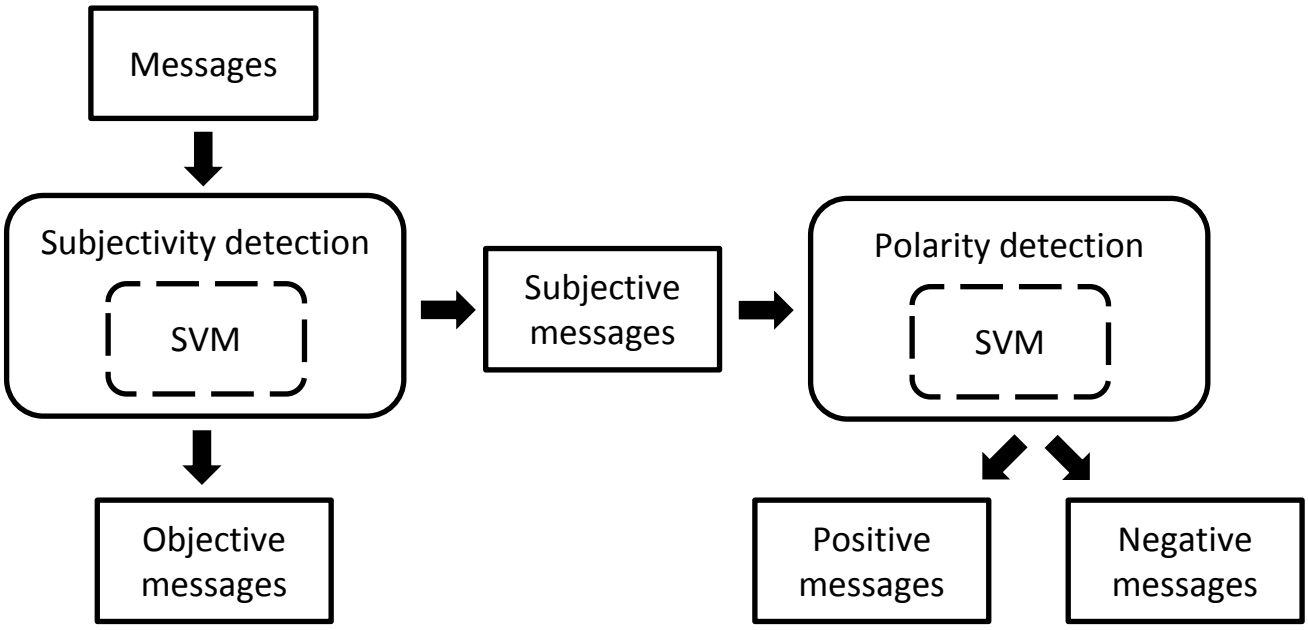


Figure 2: Our 2–stage pipeline procedure.

W_+ : Contains weak subjective expressions with positive prior polarity.

W_- : Contains weak subjective expressions with negative prior polarity.

W_0 : Contains weak subjective expressions with neutral prior polarity.

Adding to these, three more lexicons were created, one for each class (positive, negative, neutral). In particular, we employed Chi Squared feature selection (Liu and Setiono, 1995) to obtain the 100 most important tokens per class from the training set. Very few tokens were manually erased to result to the following three lexicons.

T_+ : Contains the top-94 tokens appearing in positive tweets of the training set.

T_- : Contains the top-96 tokens appearing in negative tweets of the training set.

T_0 : Contains the top-94 tokens appearing in neutral tweets of the training set.

The nine lexicons described above are used to calculate precision ($P(t,c)$), recall ($R(t,c)$) and $F-measure$ ($F_1(t,c)$) of tokens appearing in a message with respect to each class. Equations 1, 2 and 3 below provide the definitions of these metrics.

$$P(t,c) = \frac{\#\text{tweets that contain token } t \text{ and belong to class } c}{\#\text{tweets that contain token } t} \quad (1)$$

$$R(t,c) = \frac{\#\text{tweets that contain token } t \text{ and belong to class } c}{\#\text{tweets that belong to class } c} \quad (2)$$

$$F_1(t,c) = \frac{2 \cdot P(t,c) \cdot R(t,c)}{P(t,c) + R(t,c)} \quad (3)$$

3.2 Feature engineering

We employed three types of features, namely boolean features, POS based features and lexicon based features. Our goal is to build a system that is not explicitly based on the vocabulary of the training set, having therefore better generalisation capability.

3.2.1 Boolean features

Bag of words (BOW): These features indicate the existence of specific tokens in a message. We used feature selection with Info Gain to obtain the 600 most informative tokens of the training set and we then manually removed 19 of them

to result in 581 tokens. As a consequence we get 581 features that can take a value of 1 if a message contains the corresponding token and 0 otherwise.

Time and date: We observed that time and date often indicated events in the train data and such messages tend to be objective. Therefore, we added two more features to indicate if a message contains time and/or date expressions.

Character repetition: Repetitive characters are often added to words by users, in order to give emphasis or to express themselves more intensely. As a consequence they indicate subjectivity. So we added one more feature having a value of 1 if a message contains words with repeating characters and 0 otherwise.

Negation: Negation not only is a good subjectivity indicator but it also may change the polarity of a message. We therefore add 5 more features, one indicating the existence of negation, and the remaining four indicating the existence of negation that precedes (in a distance of at most 5 tokens) words from lexicons S_+, S_-, W_+ and W_-.

Hash-tags with sentiment: These features are implemented by getting all the possible substrings of the string after the symbol # and checking if any of them match with any word from S_+, S_-, W_+ and W_- (4 features). A value of 1 means that a hash-tag containing a word from the corresponding lexicon exists in a message.

3.2.2 POS based features

Specific POS tags might be good indicators of subjectivity or objectivity. For instance adjectives often express sentiment (e.g., beautiful, frustrating) while proper nouns are often reported in objective messaged. We, therefore, added 10 more features based on the following POS tags:

1. adjectives,
2. adverbs,
3. verbs,
4. nouns,
5. proper nouns,
6. urls,
7. interjections,
8. hash-tags,
9. happy emoticons, and
10. sad emoticons.

We then constructed our features as follows. For each message we counted the occurrences of tokens with these POS tags and we divided this number with the number of tokens having any of these POS tags. For instance if a message contains 2 adjectives, 1 adverb and 1 url then the features corresponding to adjectives, adverbs and urls will have a value of $\frac{2}{4}$, $\frac{1}{4}$ and $\frac{1}{4}$ respectively while all the remaining features will be 0. These features can be thought of as a way to express how much specific POS tags affect the sentiment of a message.

Going a step further we calculate precision ($P(b,c)$), recall ($R(b,c)$) and $F-measure$ ($F_1(b,c)$) of POS tags bigrams with respect to each class (equations 4, 5 and 6 respectively).

$$P(b,c) = \frac{\#\text{tweets that contain bigram } b \text{ and belong to class } c}{\#\text{tweets that contain bigram } b} \tag{4}$$

$$R(b,c) = \frac{\#\text{tweets that contain bigram } b \text{ and belong to class } c}{\#\text{tweets that belong to class } c} \tag{5}$$

$$F_1(b,c) = \frac{2 \cdot P(b,c) \cdot R(b,c)}{P(b,c) + R(b,c)} \tag{6}$$

For each bigram (e.g., adjective-noun) in a message we calculate $F_1(b,c)$ and then we use the average, the maximum and the minimum of these values to create 9 additional features. We did not experiment over measures that weight differently Precision and Recall (e.g., F_b for $b = 0.5$) or with different combinations (e.g., F_1 and P).

3.2.3 Lexicon based features

This set of features associates the words of the lexicons described earlier with the three classes. Given a message M, similarly to the equations 4 and

6 above, we calculate $P(t, c)$ and $F_1(t, c)$ for every token $t \in M$ with respect to a lexicon. We then obtain the maximum, minimum and average values of $P(t, c)$ and $F_1(t, c)$ in M. We note that the combination of P and F_1 appeared to be the best in our experiments while $R(t, c)$ was not helpful and thus was not used. Also, similarly to section 3.2.2 we did not experiment over measures that weight differently Precision and Recall (e.g., F_b for $b = 0.5$). The former metrics are calculated with three variations:

(a) **Using words:** The values of the metrics consider only the words of the message.

(b) **Using words and priors:** The same as (a) but adding to the calculated metrics a prior value. This value is calculated on the entire lexicon, and roughly speaking it is an indicator of how much we can trust L to predict class c. In cases that a token t of a message M does not appear in a lexicon L the corresponding scores of the metrics will be 0.

(c) **Using words and their POS tags:** The values of the metrics consider the words of the message along with their POS tags.

(d) **Using words, their POS tags and priors**: The same as (c) but adding to the calculated metrics an apriori value. The apriori value is calculated in a similar manner as in (b) with the difference that we consider the POS tags of the words as well.

For case (a) we calculated minimum, maximum and average values of $P(t, c)$ and $F_1(t, c)$ with respect to S_+, S_-, S_0, W_+, W_- and W_0 considering only the words of the message resulting to 108 features. Concerning case (b) we calculated average $P(t, c)$ and $F_1(t, c)$ with respect to S_+, S_-, S_0, W_+, W_- and W_0, and average $P(t, c)$ with respect to T_+, T_- and T_0 adding 45 more features. For case (c) we calculated minimum, maximum and average $P(t, c)$ with respect to S_+, S_-, S_0, W_+, W_- and W_0 (54 features), and, finally, for case (d) we calculated average $P(t, c)$ and $F_1(t, c)$ with respect to S_+, S_-, S_0, W_+, W_- and W_0 to add 36 features.

Class	F_1
Positive	0.6496
Negative	0.4429
Neutral	0.7022
Average	0.5462

Table 1: F_1 for development set.

4 Experiments

As stated earlier we use a 2–stage pipeline approach to identify the sentiment of a message. Preliminary experiments on the development data showed that this approach is better than attempting to address the problem in one stage during which a classifier must classify a message as positive, negative or neutral. To be more precise we used a Naive Bayes classifier and BOW features using both 1–stage and 2–stage approaches. Although we considered the 2–stage approach with a Naive Bayes classifier as a baseline system we used it to submit results for both twitter and sms test sets.

Having concluded to the 2–stage approach we employed for each stage an SVM classifier, fed with the 855 features described in section 3.2.[3] Both SVMs use linear kernel and are tuned in order to find the optimum C parameter. Observe that we use the same set of features in both stages and let the classifier learn the appropriate weights for each feature. During the first stage, the classifier is trained on the entire training set after merging positive and negative classes to one superclass, namely subjective. In the second stage, the classifier is trained only on positive and negative tweets of the training and is asked to determine whether the messages classified as subjective during the first stage are positive or negative.

4.1 Results

In order to obtain the best set of features we trained our system on the downloaded training data and measured its performance on the provided development data. Table 1 illustrates the F_1 results on the development set. A first observation is that there is a considerable difference between the F_1 of the negative class and the other two, with the former be-

[3] We used the LIBLINEAR distribution (Fan et al., 2008)

Class	F1
Positive	0.6854
Negative	0.4929
Neutral	0.7117
Average	0.5891

Table 2: F_1 for twitter test set.

Class	F1
Positive	0.6349
Negative	0.5131
Neutral	0.7785
Average	0.5740

Table 3: F_1 for sms test set.

ing significantly decreased. This might be due to the quite low number of negative tweets of the initial training set in comparison with the rest of the classes. Therefore, the addition of 340 negative examples from the development set emerged from this imbalance and proved to be effective as shown in table 2 illustrating our results on the test set regarding tweets. Unfortunately we were not able to submit results with this system for the sms test set. However, we performed post-experiments after the gold sms test set was released. The results shown on table 3 are similar to the ones obtained for the twitter test set which means that our model has a good generalisation ability.

5 Conclusion and future work

In this paper we presented our approach for the Message Polarity Classification task of SEMEVAL 2013. We proposed a pipeline approach to detect sentiment in two stages; first we discard objective messages and then we classify subjective (i.e., carrying sentiment) ones as positive or negative. We used SVMs with various extracted features for both stages and although the system performed reasonably well, there is still much room for improvement. A first problem that should be addressed is the difficulty in identifying negative messages. This was mainly due to small number of tweets in the training data. This was somewhat alleviated by adding the negative instances of the development data but still our system reports lower results for this class as compared to positive and neutral classes. More data or better features is a possible improvement. Another issue that has not an obvious answer is how to proceed in order to improve the 2–stage pipeline approach. Should we try and optimise each stage separately or should we optimise the second stage taking into consideration the results of the first stage?

References

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Beijing, China. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Huan Liu and Rudy Setiono. 1995. Chi2: Feature selection and discretization of numeric attributes. In *Tools with Artificial Intelligence, 1995. Proceedings., Seventh International Conference on*, pages 388–391. IEEE.

Bo Pang and Lillian Lee. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Barcelona, Spain. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *EMNLP*, pages 1524–1534.

V. Vapnik. 1998. *Statistical learning theory*. John Wiley.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005a. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354. Association for Computational Linguistics.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov, and Alan Ritter. 2013. SemEval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '13, June.