DUTH at SemEval-2017 Task 4: A Voting Classification Approach for Twitter Sentiment Analysis

Symeon Symeonidis **Dimitrios Effrosynidis** **John Kordonis** **Avi Arampatzis**

Database & Information Retrieval research unit,
Department of Electrical & Computer Engineering,
Democritus University of Thrace, Xanthi 67100, Greece
{ssymeoni,dimievfr,ioankord1,avi}@ee.duth.gr

Abstract

This report describes our participation to SemEval-2017 Task 4: Sentiment Analysis in Twitter, specifically in subtasks A, B, and C. The approach for text sentiment classification is based on a Majority Vote scheme and combined supervised machine learning methods with classical linguistic resources, including bag-of-words and sentiment lexicon features.

1 Introduction

For millions of users, microblogging services such as Twitter, a popular service where users can post no more than 140 characters status messages, have become an elemental part of daily life. By using tools and techniques from Natural Language Processing (NLP) and machine learning, Sentiment analysis is defined as the process to identify and analyze polarity from short texts, sentences, and documents (Pang et al., 2008).

In the last few years, people from different research disciplines are interested in Sentiment Analysis, and the SemEval workshop offers an opportunity to compete and work in this field. Our team has participated in SemEval-2017 task 4 on Sentiment Analysis in Twitter, more specifically on subtasks A (Message Polarity Classification), B, and C (Tweet Classification in either two-point or five-point scale respectively) (Rosenthal et al., 2017).

In this report, we present an ensemble text sentiment classification scheme, based on an extensive empirical analysis of several classifiers and other related works, e.g. (Balahur, 2013; Martínez-Cámara et al., 2014; Balikas and Amini, 2016; Onan et al., 2016). A voting scheme combines learning algorithms to identify and select an optimal set of base learning algorithms. These components were carefully combined and optimized to create a separate version of the system for each of the tackled subtasks.

The rest of this report is organized as follows. The description of proposed system we used and its feature extraction are presented in Section 2. Section 3 reports our experiments. Conclusions and directions for further work/research are summarized in Section 4.

2 System Description

The main objective of SemEval-2017 Task 4 is sentiment classification. The system we used is based on the bag-of-words representation, n-gram extraction, and usage of lexicons which have a predefined sentiment for every uni-gram and bi-gram. For the implementation of the system we used Python's Scikit-Learn (Pedregosa et al., 2011), as well as NLTK (Natural Language Toolkit) (Bird et al., 2009).

2.1 Pre-processing

The pre-processing steps that we followed were to remove and replace strings from the tweets that do not show any sentiment, as well as to remove duplicates and unicode strings:

- Removing duplicates: we found that some instances were duplicates, e.g. in Subtask A, so we removed them.
- Replacing hashtags, URLs and usernames: we first removed the "#" character in front of the words and replaced the twitter oriented strings @usernames and the URLs with tags such as "AT_USER" and "URL" respectively.
- Removing unicode strings: there were many Unicode strings especially in the testing data, e.g. strings like "\u002c" and "x96".

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 704–708,
Vancouver, Canada, August 3 - 4, 2017.

	Positive	Negative	Neutral	Total
train	18377 (38%)	7442 (16%)	22012 (46%)	47831
dev	2412 (43%)	1056 (18%)	2185 (39%)	5653
test	2375 (19%)	3972 (32%)	5937 (49%)	12284

Table 1: Number of tweets in training (train), development (dev), and testing (test) data for subtask A.

- Removing numbers and punctuation: preliminary experiments showed better results when we removed all the numbers. Before removing punctuation, we detected useful punctuation signs such as "!" and "?" and replaced them with labels.

- Using lowercase and tokenization: the final tweets were lower-cased (after detecting words that had all of their character capitalized which were retained) and splitted into tokens.

- Removing stop words: stopwords are common function words with very high frequency among sentences and low content, so we removed them.

- Using stemming: stemming is the process of reducing a word to its base root form. Preliminary tests showed that stemming improves a lot the results.

Previous studies (Pak and Paroubek, 2010; Bakliwal et al., 2013) have made references on the influence of pre-processing and proposed a set of features to extract the maximum sentimental information.

2.2 Feature Engineering

We extracted features based on the lexical content of each tweet and we also used lexicons. Below we present all the features.

- Word n-grams: the word level uni-grams and bi-grams are adopted.

- Number of capitalized words

- Number of question marks, exclamation marks and the aggregation of them

- Number of elongated words: it indicates the number of elongated words in the raw text of the tweet.

To identify the sentiment polarity of tweets, we used three different sentiment lexicons during our experiments. Sentiment lexicons are lexical resources which are formed by a list of words without any additional information and are built by opinion words and some sentiment phrases (Martínez-Cámara et al., 2014).

In our system we used sentiment lexicons such as Bing Liu's lexicon (Hu and Liu, 2004), the NRC emotion lexicon (Mohammad and Turney, 2010), the MPQA lexicon (Wilson et al., 2005) and combinations of them. The above lexicons have a sentiment tag for each word and in our approach we count the occurrences of each sentiment class for each tweet's word. Finally, we compute the overall sentiment of the tweet, by adding its words sentiments.

3 Experiments

In this section, after the feature extraction, we analyse the classification process with the learning methods and classification algorithms that used in our system.

3.1 Datasets

The datasets were provided by the organizers and contained all datasets of the previous years with the addition of a new. For Subtask A the available datasets were all the training, development, and testing data from the years 2013 to 2016. For Subtask B the available datasets were from the years 2015 to 2016, and for Subtask C from the year 2016. We used a portion of the data for development and the rest for training. We present them in Tables 1–3.

	Positive	Negative	Total
train	12812 (79%)	3410 (21%)	16222
dev	2139 (78%)	604 (22%)	2743
test	2463 (40%)	3722 (60%)	6185

Table 2: Number of tweets in training (train), development (dev), testing (test) data for subtask B.

	2	1	0	-1	-2	Total
train	819 (3%)	10984 (41%)	11735 (44%)	2869 (11%)	225 (1%)	26632
dev	201 (5%)	1938 (48%)	1258 (31%)	529 (13%)	74 (3%)	4000
test	131 (1%)	2332 (19%)	6194 (50%)	3545 (29%)	177 (1%)	12379

Table 3: Number of tweets in training (train), development (dev), testing (test) data for subtask C.

As we can observe from the tables, the testing data that were provided by the organizers have different ratio among the classes, especially between the positives and negatives.

3.2 Evaluation Metrics

For Subtask A, we use the macro-average recall, which is the recall averaged across the three classes $R_{macro} = \frac{R_{pos}+R_{neu}+R_{neg}}{3}$. Subtask B maintains the same measure, but among the two classes $R_{macro} = \frac{R_{pos}+R_{neg}}{2}$. For Subtask C, the official metrics are the macro-averaged mean absolute error and the extension of macro-averaged recall for ordinal regression (Rosenthal et al., 2017) among 5 predefined classes.

3.3 Learning

Using all the features described above, we first trained several classifiers to the development data in order to tune the parameters of each classifier. The main target of tuning was the metric of this specific task, which is the macro-average recall. We tested a variety of classifiers that include the following:

- Ridge: an algorithm belonging to the Generalized Linear Models family that alleviates the multicollinearity amongst predictor variables.
- Logistic Regression: despite its name it is used for classification and fits a linear model. It is also known as Maximum Entropy, and uses a logistic function to model the probabilities that describe the output prediction.
- Stochastic Gradient Descent: a simple and efficient algorithm to fit linear models. It is suitable for very large number of features.
- Nearest Centroid: an algorithm that uses the center of a class, called centroid, to represent it and has no parameters.
- Bernoulli Naïve Bayes: an alternative of Naïve Bayes, where each term is equal to 1 if it exists in the sentence and 0 if not. Its difference from Boolean Naïve Bayes is that it takes into account terms that do not appear in the sentence.
- Linear SVC: an SVM algorithm, which tries to find a set of hyperplanes that separate space into dimensions representing classes. The hyperplanes are chosen in a way to maximize the distance from the nearest data point of each class.
- Passive-Aggressive: belongs to a family of algorithms for large-scale learning, which do not require a learning rate and includes a regularization parameter C (Pedregosa et al., 2011).

In order to vectorize the collection of raw documents, we used a Python's Scikit-Learn (Pedregosa et al., 2011) *tf-idf* transformation with a *max_df* parameter of 0.5. The value of this parameter was extracted by the tuning process and indicates that we ignore terms that have a frequency strictly higher than this threshold. The next step was to use these parameters to test our model with the help of 10-fold cross-validation on the training set.

3.3.1 Subtask A

Subtask A is a multi-class classification problem, where each tweet has to be classified in one among three classes. We found that the best combination for this task was the use of stemming and the three lexicons. Features like the number of exclamation marks, etc., under-performed. The three classifiers with the best results were the Bernoulli Naïve Bayes, the Stochastic Gradient Descent (SGD), and the Linear SVC.

The final step was to use the majority voting classification method that combines three different classifiers and outputs the class that the majority of them agreed. Using all possible combinations of every three classifiers, the best result was with the Bernoulli Naïve Bayes, SGD, and Nearest Centroid. Note that Nearest Centroid was one of the

weakest classifiers in isolation, but presented an excellent contribution when combined with other two.

3.3.2 Subtask B

Subtask B is a topic-based binary classification problem, where each tweet belongs to a topic, and one has to classify whether the tweet conveys a positive or negative sentiment towards the topic. We used the same approach with Subtask A, with the addition of a weight for the topic which was added as a feature. The best combination was the use of stemming and the three lexicons, like in subtask A. The three best classifiers were the SGD, the Passive-Aggressive, and the Linear SVC.

The majority voting classifier outperformed all the single classifiers; here, the best result was with the SGD, Logistic Regression, and Ridge classifiers, showing once again that weak classifiers can contribute significantly when combined with others.

3.3.3 Subtask C

Subtask C is also a topic-based classification problem, where each tweet belongs to a topic, and one has to estimate the sentiment conveyed by the tweet towards the topic on a five-point scale. The same approach as with Subtask B was used, and the best result was achieved by the combination of the Logistic Regression, the Nearest Centroid, and the Bernoulli Naïve Bayes classifiers.

	ρ	F_1^{PN}	Acc
Task A	0.621	0.605	0.640
Task B	0.663	0.600	0.607
	(MAE^M)	(MAE^μ)	
Task C	0.895	0.544	

Table 4: DUTH's results for SemEval-2017 Task 4 on Sentiment Analysis in Twitter (Rosenthal et al., 2017).

4 Conclusions & Future work

By analyzing and classifying sentiments on Twitter, people can comprehend attitudes about particular topics, making Sentiment Analysis an attractive research area. In this report we presented an approach for Twitter sentiment analysis on two-point, three-point, and five-point scale, based on a voting classification method. This was our first contact with the task of sentiment analysis and compared with the top-ranked participating systems, there seems to be for us much room for improvement.

In future work, we consider to focus on adding more pre-processing methods such as spelling correction and POS tagging. We also consider adding more features such as emoticons, negation, character n-grams and more lexicons.

References

Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment analysis of political tweets: Towards an accurate classifier. Association for Computational Linguistics.

Alexandra Balahur. 2013. Sentiment analysis in social media texts. In *4th workshop on computational approaches to subjectivity, sentiment and social media analysis*. pages 120–128.

Georgios Balikas and Massih-Reza Amini. 2016. Twise at semeval-2016 task 4: Twitter sentiment classification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 85–91. http://www.aclweb.org/anthology/S16-1010.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '04, pages 168–177. https://doi.org/10.1145/1014052.1014073.

Eugenio Martínez-Cámara, Salud María Jiménez-Zafra, Maite Martin, and L. Alfonso Urena Lopez. 2014. Sinai: Voting system for twitter sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 572–577. http://www.aclweb.org/anthology/S14-2100.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, Stroudsburg, PA, USA, CAAGET '10, pages 26–34. http://dl.acm.org/citation.cfm?id=1860631.1860635.

Aytu Onan, Serdar Korukolu, and Hasan Bulut. 2016. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications* 62:1 – 16.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*. volume 10.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12:2825–2830. http://dl.acm.org/citation.cfm?id=1953048.2078195.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 501–516. http://www.aclweb.org/anthology/S17-2088.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '05, pages 347–354. https://doi.org/10.3115/1220575.1220619.