senti.ue-en: an approach for informally written short texts in SemEval-2013 Sentiment Analysis task

José Saias
DI - ECT - Universidade de Évora
Rua Romão Ramalho, 59
7000-671 Évora, Portugal
jsaias@uevora.pt

Hilário Fernandes
Cortex Intelligence
Rua Sebastião Mendes Bolas, 2 K
7005-872 Évora, Portugal
hilario.fernandes@cortex-intelligence.com

Abstract

This article describes a Sentiment Analysis (SA) system named `senti.ue-en`, built for participation in SemEval-2013 Task 2, a Twitter SA challenge. In both challenge subtasks we used the same supervised machine learning approach, including two classifiers in pipeline, with 22 semantic oriented features, such as polarized term presence and index, and negation presence. Our system achieved a better score on Task A (0.7413) than in the Task B (0.4785). In the first subtask, there is a better result for SMS than the obtained for the more trained type of data, the tweets.

1 Introduction

This paper describes the participation of a group led by Universidade de Évora's Computer Science Department in SemEval-2013 Task 2 (Wilson et al., 2013), using `senti.ue-en` system. Having previous experience in NLP tasks, such as question answering (Saias, 2010; Saias and Quaresma, 2012), this was the authors first attempt to implement a system for Sentiment Analysis (SA) in English language. We have a recent work (Fernandes, 2013) involving SA but it is geared towards Portuguese language, and thought for regular text. It was based on rules on the outcome of linguistic analysis, which did not work well for tweets, because the morphosyntactic analyzer misses much, due to the abundance of writing errors, symbols and abbreviations. Moreover, in that work we began by detecting named entities and afterwards classify the sentiment being expressed about them. For SemEval the goal is different, being target-independent. In both A and B subtasks, systems must work on sentiment polarity, in a certain context or full message, but the target entity (or the opinion topic) will not appear in the output. Thus, we have decided that `senti.ue-en` system would be implemented from scratch, for English language and according to the objectives of this challenge, in particular the Task B.

2 Related Work

Microblogging and social networks are platforms where people express opinions. In recent years many papers have been published on social media content SA. Pang et al. (2002) applied machine learning based classifiers for sentiment classification on movie reviews. Their experimental results using Naive Bayes, Maximum Entropy, and Support Vector Machines (SVM) algorithms achieved best results with SVM and unigram presence as features.
Some target-dependent approaches are sensitive to the entity that is receiving each sentiment. A sentence can have a positive sentiment about an entity and a negative for another. Such classification can be performed with rules on the occurrence of nouns, verbs and adjectives, as done in (Nasukawa and Yi, 2003). It is common to use parsers and part-of-speech tagging. Barbosa and Feng (2010) explore tweet writting details and meta-information in feature selection. Instead of using many unigrams as features, the authors propose the use of 20 features (related to POS tags, emoticons, upper case usage, word polarity and negation), achieving faster training and test times. A two-phase approach first clas-

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 508–512, Atlanta, Georgia, June 14-15, 2013.

sifies messages as subjective and objective, and then the polarity is classified as positive or negative for tweets having subjectivity. Groot (2012) builds a feature vector with polarized words and frequently occurring words being taken as predictive for Twitter messages. Supervised learning algorithms as SVM and Naive Bayes are then used to create a prediction model. The work (Gebremeskel, 2011) is focused on tweets about news. Authors report an accuracy of 87.78% for a three-classed sentiment classification using unigram+bigram presence features and Multinomial Naive Bayes classifier. In Jiang et al. (2011) work, Twitter SA starts with a query, identifying a target, and classifies sentiment in the query result tweets, related to that target. Instead of considering only the text of a tweet, their context-aware approach also considers related tweets and target-dependent features. With precise criteria for the context of a tweet, authors seek to reduce ambiguity and report performance gains.

3 Methodology

As in most systems described in the literature, in this area, our `senti.ue-en` system is based on supervised machine learning. To handle the data format, in the input and on the outcome of the system, we chose to use Python and the Natural Language Toolkit (NLTK), a platform with resources and programming libraries suitable for linguistic processing (Bird, 2006). Task A asks us to classify the sentiment in a word or phrase in the context of the message to which it belongs. For Task B, we had to classify the overall sentiment expressed in each message. Since tweets are short messages, we early have chosen to apply the same system for both tasks, admitting some possible difference in training or parameterization. As the fine control of the correspondence between each sentiment expression and its target entity is not sought, Task A is treated as a special case of Task B, and our system does not consider the text around the expression to classify. The organization prepared a message corpus for training and another to be used as a development-time evaluation dataset. We merged the training corpus with the development corpus, and our development test set was dynamically formed by random selection of instances for each class (positive, negative and neutral). Some tweets were not downloaded properly. For message polarity classification, we ended up with 9191 labeled messages, which we split into training and test sets.

Text processing started with tokenization, that was white space or punctuation based. Some experiments also included lemmatization, done with the NLTK WordNet Lemmatizer. In the first approach to Task B, we applied the Naive Bayes classification algorithm using term presence features. The test set was formed by random selection of 200 instances of each class. After several experiments with this system configuration, the average accuracy for the 3 classes was close to 45%. Looking for better results, instead of the bag-of-words approach, we chose a smaller set of semantic oriented features:

- presence of polarized term
- overall value of sentiment in text
- negation presence
- negation before polarized expression
- presence of polarized task A n-grams
- overall value of polarized task A n-grams
- overall and presence of similar to Task A n-grams
- first and last index of polarized terms

Checking for the presence of positive and negative polarized terms produces two features for each of the three sentiment lexicons used by our system. AFINN (Nielsen, 2011) is a sentiment lexicon containing a list of English words rated between minus five (negative) and plus five (positive). The words have been manually labeled by Finn Årup Nielsen, from 2009 to 2011. SentiWordNet (Baccianella et al., 2010) is a lexical resource for opinion mining that assigns sentiment scores to each synset of WordNet (Princeton University, 2010). After some experimentation with this resource, we decided to apply a threshold, disregarding terms whose score absolute value is less than 0.3. Another sentiment lexicon, from Liu et al. (2005), derived from a work on online customer reviews of products. The overall text sentiment value is calculated by adding the sentiment value in each word. This is the way chosen to handle more than one sentiment in a single tweet. Our system creates a separated overall sentiment value feature for AFINN, SentiWordNet and Liu's lexicons, because each resource uses a different range of values. Each of these features is calculated by summing the sentiment value in each word of the text classify. Detection of denial in the text also gave rise to a feature. Thinking in cases like "*This meal was not*

good", we created features for the presence of denial before positive and negative expressions, where the adjective's sentiment value is inverted by negation. In these two features, an expression is polarized if it is included in any of the sentiment lexicons. The training corpus for Task A included words or phrases marked as positive or negative. We created two more features to signal the presence of polarized words or n-grams in the texts to be classified. To complement, another feature accounts for the overall Task A polarized n-grams value, adding 1 for each positive occurrence and subtracting 1 every negative occurrence in the tweet. Because a term can arise in inflected form, we added another three features to assess the same on Task A data, but accepting variations in words or expressions. Using lemmatization and synonyms, we seek more flexibility in n-gram verification. The last four features identify the text token index for the first and the last occurrence, for each sentiment flavor, positive and negative, according to any used sentiment lexicon. Emoticons are present in sentiment lexicons, so it was not created a specific feature for them.

Using these 22 features with Naive Bayes, the average overall accuracy was 60%. When analyzed by class, the lower accuracy happens on neutral class, near 50%. Accuracy por positive class was 68%, and for negative it was 63%. For the next iteration, the NLTK classifier was set up for Decision Tree algorithm. After several runs, we noticed that while the overall accuracy remained identical, the poorest results came now for the negative class, having 54% accuracy. The run average accuracy for classes positive and neutral, was respectively 59% and 64%. In the latest evolution the system applies two classifiers in sequence. Each tweet is first classified with Naive Bayes. This creates a new feature for the second classifier, which is considered along with the previous ones by the Decision Tree algorithm. This configuration led us to the best overall accuracy in the development stage, with 62%, and was the version applied to Task B in constrained mode.

The unconstrained mode allowed systems to use additional data for training. The IMDB dataset (Maas et al., 2011) contains movie reviews with their associated binary sentiment polarity labels. We chose a subset of this corpus consisting of 500 positive and 500 negative reviews with less than 350 characters.

T	Data	Mode	Positive	Negative	Neutral
A	sms	C	0.8079	0.8985	0.1130
		U	0.8695	0.9206	0.1348
	twitter	C	0.9190	0.8162	0.0588
		U	0.9412	0.8411	0.0705
B	sms	C	0.4676	0.4356	0.7168
		U	0.4625	0.4161	0.7293
	twitter	C	0.6264	0.3996	0.5538
		U	0.6036	0.3589	0.5621

Table 1: `senti.ue-en` precision in Tasks A and B

Sanders used a Naive Bayes classifier and token-based feature extraction to create a corpus (Sanders, 2011) for SA on Twitter. We were able to discharge only part of the corpus, from which we selected 250 positive tweets and the same number of negative ones. In unconstrained mode, `senti.ue-en` has the same configuration, but uses extra instances from these two corpus for training.

Task A is treated with the same mechanism. The system classifies the sentiment for the text inside the given boundaries. Because many of these cases have a single word, our system uses a third extra corpus for training in unconstrained mode. Each word on AFINN lexicon is added to training set, with positive or negative class, depending on its sentiment value.

4 Results

We submitted our system's result for each of the eight expected runs. Each run was a combination of subtask (A or B), dataset (Twitter or SMS) and training mode (constrained or unconstrained). After the deadline for submission, the organization evaluated the results. The precision in our system's output is indicated in Table 1. The use of more training instances in unconstrained mode leads to an improvement of precision in Task A, for all classes. In Task B we notice the opposite effect, with a slight drop in precision for positive and negative classes, and about 1% improvement in neutral class precision. We also note that precision has lower values in neutral class for Task A, whereas in Task B it is the class negative that has the lowest precision.

Table 2 shows the recall obtained for the same results. This metric also shows a gain in Task A, for positive and negative classes using unconstrained mode. For subtask B, the constrained mode had bet-

T	Data	Mode	Positive	Negative	Neutral
A	sms	C	0.5341	0.5453	0.6792
		U	0.6471	0.6196	0.6730
	twitter	C	0.4898	0.4958	0.7500
		U	0.6203	0.5704	0.7000
B	sms	C	0.5711	0.3350	0.7061
		U	0.5386	0.4594	0.6556
	twitter	C	0.5515	0.3245	0.6555
		U	0.5280	0.4359	0.5854

Table 2: `senti.ue-en` recall in Tasks A and B

T	Data	Mode	Positive	Negative	Neutral
A	sms	C	0.6431	0.6787	0.1937
		U	0.7420	0.7407	0.2246
	twitter	C	0.6390	0.6169	0.1090
		U	0.7478	0.6798	0.1281
B	sms	C	0.5142	0.3788	0.7114
		U	0.4977	0.4367	0.6905
	twitter	C	0.5866	0.3581	0.6004
		U	0.5633	0.3937	0.5735

Table 3: `senti.ue-en` F-measure in Tasks A and B

ter recall for positive and neutral classes. But recall varies in the opposite direction in the negative class when using our extra training instances.

Using the F-measure metric to evaluate our results, we get the values in Table 3. This balanced assessment between precision and recall confirms the improvement of results in Task A when using the unconstrained mode. We note, for Task B, a small loss in unconstrained mode on positive class, but that is outweighed by the gain on the negative class.

In SemEval-2013 Task 2, the participating systems are ranked by their score. This corresponds to the average F-measure in positive and negative classes. Table 4 shows the score obtained by our system. The score is in line with our forecasts in the Task A, but below what we wanted in Task B. Looking at Table 3 we see that positive and negative classes' F-measure values are substantially lower than the values for neutral class, in Task B and in both constrained and unconstrained mode. For Task B, most correct results were in the class less relevant for the score.

5 Conclusions

With our participation in SemEval-2013 Task 2 we intended to build a real-time SA system for the English used nowadays in social media content. This goal was achieved and we experienced the use of important English linguistic resources to support this task, such as corpora and sentiment lexicons.

T	Data	Mode	Score
A	sms	C	0.6609
		U	0.7413
	twitter	C	0.6279
		U	0.7138
B	sms	C	0.4465
		U	0.4672
	twitter	C	0.4724
		U	0.4785

Table 4: `senti.ue-en` score

We had some problems detected only after the close of submission. Lemmatization did not always work well. In 'last index of polarized term' feature, we noticed a problem that ironically came precisely at the version used to submit, where the last index was counted from the start of text, and it should be counted from the end.

We think that the difference in system performance between Task A and Task B has to do with the amount of noise present in the text. Because many of the texts to classify in Task A had a single word or a short phrase, the system was more likely to succeed. Another reason is the fact that our system has not been tuned to maximize the score (F-measure in positive and negative classes). During development we took into account only the overall accuracy seen in NLTK classifier result. Perhaps the overall system performance may have been affected by our decision of merge the training and the development corpus as training set. We used a class balanced set for development-time evaluation, smaller than the given development set, and the final test set had a different class distribution (Wilson et al., 2013).

By reviewing the system, we feel that the classification algorithms in the pipeline system should swap. Now we would use first the Decision Tree classifier, and after, receiving an extra feature, the Naive Bayes classifier, which as mentioned in section 3, suggested slightly better results for positive and negative classes. For the future, we intend to evolve the system in order to become more precise and target-aware. For the first part we need to review and evaluate the actual contribution of the current features. As for the second, we intend to introduce named entity recognition, so that each sentiment can be associated with its target entity.

References

Andrew L. Maas, Raymond Daly, Peter Pham, Dan Huang, Andrew Ng and Christopher Potts. 2011. *Learning Word Vectors for Sentiment Analysis*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp: 142-150. ACL. Portland, USA.

Bing Liu, Minqing Hu and Junsheng Cheng. 2005. *Opinion Observer: Analyzing and Comparing Opinions on the Web*. In Proceedings of the 14th International World Wide Web conference (WWW-2005). Chiba, Japan.

Bo Pang, Lillian Lee, Shivakumar Vaithyanathan. 2002. *Thumbs up? Sentiment Classification using Machine Learning Techniques*. In Proceedings of EMNLP. pp: 79–86.

Finn Årup Nielsen. 2011. *A New ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs*. In Proceedings, 1st Workshop on Making Sense of Microposts (#MSM2011): Big things come in small packages. pp: 93-98. Greece. http://arxiv.org/abs/1103.2903

Gebrekirstos Gebremeskel. 2011. *Sentiment Analysis of Twitter Posts About news*. Master's thesis. University of Malta.

Hilário Fernandes. 2013. *Sentiment Detection and Classification in Non Structured Information Sources*. Master's thesis, ECT - Universidade de Évora.

José Saias. 2010. *Contextualização e Activação Semântica na Selecção de Resultados em Sistemas de Pergunta-Resposta*. PhD thesis, Universidade de Évora.

José Saias and Paulo Quaresma. 2012. Di@ue in clef2012: question answering approach to the multiple choice qa4mre challenge. In *Proceedings of CLEF 2012 Evaluation Labs and Workshop - Working Notes Papers*, Rome, Italy.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 151–160. Association for Computational Linguistics. USA.

Luciano Barbosa and Junlan Feng. 2010. *Robust Sentiment Detection on Twitter from Biased and Noisy Data*. Coling 2010. pages 36–44. Beijing.

Niek J. Sanders. 2011. *Sanders-Twitter Sentiment Corpus*. Sanders Analytics LLC

Princeton University. 2010. "About WordNet." WordNet. http://wordnet.princeton.edu

Roy de Groot. 2012. *Data mining for tweet sentiment classification*. Master's thesis, Faculty of Science - Utrecht University.

Stefano Baccianella, Andrea Esuli and Fabrizio Sebastiani. 2010. *SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining*. In Proceedings of the Seventh conference on International Language Resources and Evaluation - LREC'10. European Language Resources Association. Malta.

Steven Bird. 2006. *NLTK: the natural language toolkit*. In Proceedings of the COLING'06/ACL on Interactive presentation sessions. Australia. http://nltk.org

Tetsuya Nasukawa, Jeonghee Yi. 2003. *Sentiment analysis: capturing favorability using natural language processing*. In Proceedings of the 2nd International Conference on Knowledge Capture(K-CAP). USA.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal and Veselin Stoyanov. 2013. SemEval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computation Linguistics.