Senti.ue: Tweet Overall Sentiment Classification Approach for SemEval-2014 Task 9

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

Abstract

This document describes the senti.ue system and how it was used for participation in SemEval-2014 Task 9 challenge. Our system is an evolution of our prior work, also used in last year's edition of Sentiment Analysis in Twitter. This system maintains a supervised machine learning approach to classify the tweet overall sentiment, but with a change in the used features and the algorithm. We use a restricted set of 47 features in subtask B and 31 features in subtask A.
In the constrained mode, and for the five data sources, senti.ue achieved a score between 78,72 and 84,05 in subtask A, and a score between 55,31 and 71,39 in subtask B. For the unconstrained mode, our score was slightly below, except for one case in subtask A.

1 Introduction

This paper describes the approach taken by a team of Universidade de Évora's Computer Science Department in SemEval-2014 Task 9: Sentiment Analysis in Twitter (Rosenthal et al., 2014). SemEval-2014 Task 9 has an expression-level (subtask A) and a message-level (subtask B) polarity classification challenges. The first subtask aims to determine whether a word (or phrase) is positive, negative or neutral, within the textual context in which it appears. The second subtask concerns the classification of the overall text polarity, which corresponds to automatically detecting the sentiment expressed in a Twitter message. In both subtasks, systems can operate in constrained or unconstrained mode. Constrained means that learning is based only on provided training texts, with the possible aid of static resources such as lexicons. Extra tweets or additional annotated documents for training are permitted only in unconstrained mode.
The system we used to respond to this challenge is called senti.ue, and follows on from our previous work on Natural Language Processing (NLP) and Sentiment Analysis (SA). We developed work in automatic reputation assessment, using a Machine Learning (ML) based classifier for comments with impact on a particular target entity (Saias, 2013). We also participated in the previous edition of SemEval SA task, where we have implemented the basis for the current system. In last year's solution (Saias and Fernandes, 2013), we treated both subtasks using the same method (except the training set). We have updated the method for subtask A, now considering also the text around the area to classify, by dedicating new features to those preceding and following tweet parts. Text overall sentiment classification is the core objective of our system, and is performed, as before, with a supervised machine learning technique. For subtask B, we fixed some implementation issues in the previous version, and we went from 22 to 53 features, explained in Section 3.

2 Related Work

The popularity of social networks and microblogging facilitated the sharing of opinions. To know whether people are satisfied or not with a particular brand or product is of great interest to marketing companies. Much work has appeared in SA, trying to capture valuable information in expressions of contentment or discontentment.
Important international scientific events, NLP related, include SA challenges and workshops. This was the case in SemEval-2013, whose task 2 (Wilson et al., 2013) required sentiment analysis of Twitter and SMS text messages. Being the pre-

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 546–550,
Dublin, Ireland, August 23-24, 2014.

decessor task of the challenge for which this work was developed, it is similar to this year's Task 9. The participating systems achieved better results in contextual polarity subtask (A) than those obtained for the overall message polarity subtask (B). In that edition, the best results were obtained by systems in constrained mode. The most common method was supervised ML with features that can be related to text words, syntactic function, discourse elements relation, internet slang and symbols, or clues from sentiment lexicons. In that task, the NRC-Canada system (Mohammad et al., 2013) obtained the best performance, achieving an F1 of 88.9% in subtask A and 69% in subtask B. That system used one SVM classifier for each subtask, together with text surface based features, features associated with manually created and automatically generated sentiment lexicons, and n-gram features. Other systems with good results in that task were GU-MLT-LT (Günther and Furrer, 2013) and AVAYA (Becker et al., 2013). The first was implemented in the Python language. It includes features for: text tokens after normalization, stems, word clusters, and two values for the accumulated positive and accumulated negative SentiWordNet (Baccianella et al., 2010) scores, considering negation. Its machine learning classifier is based on linear models with stochastic gradient descent. The approach taken in the AVAYA system centers on training high-dimensional, linear classifiers with a combination of lexical and syntactic features. This system uses Bag-of-Words features, with negation represented in word suffix, and including not only the raw word forms but also combinations with lemmas and PoS tags. Then, word polarity features are added, using the MPQA lexicon (Wiebe et al., 2005), as well as syntactic dependency and PoS tag features. Other features consider emoticons, capitalization, character repetition, and emphasis characters, such as asterisks and dashes. The resulting model was trained with the LIBLINEAR (Fan et al., 2008) classification library.

Another NLP task very close to SA is polarity classification on the reputation of an entity. Here, instead the sentiment in the perspective of the opinion holder, the goal is to detect the impact of this particular opinion on some entity's reputation. The diue system (Saias, 2013) uses a supervised ML approach for reputation polarity classification, including Bag-of-Words and a limited set of features based on sentiment lexicons and superficial text analysis.

3 Method

This work follows on from our previous participation in SemEval-2013 SA task, where we have devoted greater effort to subtask B. We start by explaining our current approach for this subtask, and then we describe how such classifier is also used in subtask A.

3.1 Message Polarity Classification

The senti.ue system maintains a supervised machine learning approach to perform the overall sentiment classification. As before, Python and the Natural Language Toolkit (NLTK[1]) are used for text processing and ML feature extraction.

The first step was to obtain the tweet content and forming the instances of the training set. During the download phase, several tweets were not found. In constrained mode, we got only 7352 instances available for training.

Tweet preprocessing includes tokenization, which is punctuation and white space based, negation detection, and lemmatization, through NLTK class WordNetLemmatizer. After that, the system runs the ML component. Instead of the solution we used in 2013, with two differently configured classifiers in a pipeline, we chose to use a single classifier, which this year is based on SciKit-Learn[2], and to increase the number of features that are extracted to represent each instance. The classification algorithm was Support Vector Machines (SVM), using SVC[3] class, with a linear kernel and 10^{-5} tolerance for stopping criterion. SVC class implementation is based on libsvm (Chang and Lin, 2011), and uses one-against-one approach for multi-class classification. From each instance, the system extracts the 47 features in Figure 1. The first two features represent the index of the first polarized token. The following represent the repeated occurrence of a question mark, and the existence of a token with negation (*not*, *never*). Then there are two features that indicate whether there is negation before positive or negative words. The following 8 fea-

[1] Python based platform with resources and programming libraries suitable for linguistic processing (Bird, 2006).

[2] Open source tool for Machine Learning in Python - http://scikit-learn.org/

[3] http://scikit-learn.org/stable/modules/svm.html#svm-classification

tures indicate whether there are positive or negative terms, just after, or near, a question mark or an exclamation mark. We build a table with words or phrases marked as positive or negative in subtask A data. Using this resource, 4 features test the presence and the count of word n-grams marked as positive or negative. Then the *TA.alike* features represent the same, but after lemmatization and synonym verification. To find the synonyms of a term, we used the WordNet (Princeton University, 2010) resource. The probability of each word belonging to a class was calculated. There are 3 features *avgProbWordOn*, one per class, that represent the average of this probability for each instance words. Next 3 features represent the same, but focusing only on the last 5 words of each text. Then we have 6 *ProbLog2Prob* features, representing the average of $P \times \log_2(P)$, for all words, or only the latest 5 words, for all classes. P is the probability of the word belonging to one class. One feature cumulates the token polarity values, according to SentiWordNet. The final 12 features are based on sentiment lexicons: AFINN (Nielsen, 2011), Bing Liu (Liu et al., 2005), MPQA, and a custom polarity table with some manually entered entries. For each resource, we count the instance tokens with negative and positive polarity, and create a feature *direction*, having the value 1 if *countTokens.pos*>*countTokens.neg*, -1 if *countTokens.pos*<*countTokens.neg*, or 0.

For the unconstrained mode, the only difference is the use of more instances for the training set, with 3296 short texts obtained from SemEval-2014 Task 4 data[4], about laptops and restaurants.

```
firstIndexOf.{pos,neg}, hasDoubleQuestionMark,
hasNegation, hasNegationBefore.{pos,neg},
{pos,neg}.{After,Near}.Exclamation,
{pos,neg}.{After,Near}.Question,
hasTA.{pos,neg}.NGrams, countTA.{pos,neg}.NGrams,
hasTA.alike.{pos,neg}.NGrams,
countTA.alike.{pos,neg}.NGrams,
avgProbWordOn.{pos,neg,neutral},
last5AvgProbWordOn.{pos,neg,neutral},
avgW.ProbLog2Prob.{pos,neg,neutral},
last5AvgW.ProbLog2Prob.{pos,neg,neutral},
SentiWordNetAccumulatedValue,
{AFINN,Liu,MPQA,custom}.countTokens.{pos,neg},
{AFINN,Liu,MPQA,custom}.direction
```

Figure 1: features for message polarity

```
{AFINN,Liu,custom}.countTokens.{pos,neg},
{AFINN,Liu,custom}.direction,
{AFINN,Liu,custom}.sentence.countTok.{pos,neg},
{AFINN,Liu,custom}.sentence.direction,
{AFINN,Liu,custom}.left.countTokens.{pos,neg},
{AFINN,Liu,custom}.left.direction,
b.sentClass.{left,right,sentence,chosenText}
```

Figure 2: features for contextual polarity

3.2 Contextual Polarity Disambiguation

In this subtask, the download phase fetched only 6506 tweets. These instances have boundaries marking the substring to classify. Our system starts by splitting the document into text segments: *fullText, leftText, rightText, sentenceText, chosenText.* The first corresponds to the entire tweet. The following represent the text before and the text after the chosen text. Then we have the sentence where the chosen text is, and finally the text segment that systems must classify. The preprocessing described before is then applied to each of these text segments. For each instance, the system generates the 31 features listed in Figure 2. First 27 features represent 9 values for each *chosenText*, *sentenceText* and *leftText* instance segments. These values represent the count of polarized tokens, and the direction (1, 0, or -1, as before), according to 3 sentiment lexicons. The final 4 features have the overall sentiment classification, using the subtask B classifier, for each text segment: *leftText*, *rightText*, *sentenceText*, and *chosenText*. In unconstrained mode the instances used for subtask A training are the same. The difference with respect to the constrained mode is the overall sentiment classifier used for the last 4 features, which corresponds to the unconstrained classifier of subtask B.

This subtask has specific features, different from those used in the previous subtask, and after some tests with SciKit-Learn classifiers, we found that, in this case, our best results were not obtained with SVM. For subtask A, we chose Gradient Boosting classifier[5], an ensemble method that combines the predictions of several models, configured with deviance loss function, 0.1 for learning rate, and 100 regression estimators with individual maximum depth of 4.

[4] http://alt.qcri.org/semeval2014/task4/

[5] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

run	LJ'14	SMS'13	T'13	T'14	T'14s
A const.	81,90	78,72	84,05	80,54	82,75
A unc.	79,70	82,93	83,80	77,07	80,02
B const.	71,39	59,34	67,34	63,81	55,31
B unc.	68,08	56,16	65,21	61,47	54,09

Table 1: `senti.ue` score

	LJ'14	SMS'13	T'13	T'14	T'14s
A avg	77,08	77,37	79,94	76,84	68,33
A best	85,61	89,31	90,14	86,63	82,75
B avg	63,52	55,63	59,78	60,41	45,44
B best	74,84	70,28	72,12	70,96	58,16

Table 2: all systems: higher and average score

4 Results

We submitted four runs, with the system output for each subtask, and both constrained and unconstrained modes. Test set documents come from five sources: LiveJournal blogs (LJ'14), SMS test (SMS'13) and Twitter test (T'13) data from last year, a new Twitter collection (T'14), and 100 tweets whose text includes sarcasm (T'14s). The primary metric to evaluate the results is the average F-measure for positive and negative classes. Table 1 shows the score obtained by our system. In the constrained mode, and for the five data sources, `senti.ue` achieved a score between 78,72 and 84,05 in subtask A, and a score between 55,31 and 71,39 in subtask B. Comparing the evaluation between constrained and unconstrained modes, the latter was always a little below, except for one case in subtask A and SMS2013 data, where the extra training data led to a 4% score improvement. In this SA challenge there were a total of 27 submissions in subtask A and 50 submissions in subtask B. Among these, the best score and the average score for each subtask are shown in Table 2. In both subtaks, our system result is above the participating systems average score. In subtask A and the Twitter Sarcasm 2014 collection (T'14s), `senti.ue` achieved the highest score, with 82,75% in constrained mode.

For each data set, tables 3 and 4 show the precision and recall of our system result on the highest scored mode, per class. In subtask A precision is between 64 and 99% for positive and negative classes, taking the value of zero in the neutral class. For the overall sentiment subtask, precision is similar among the 3 classes, having the minimum value in the negative class of sarcasm tweets. The best recall value was obtained in the positive class of the 2014 tweet collection.

task, mode, data	Positive	Negative	Neutral
A, C, LJ'14	87,27	86,69	0,00
A, U, SMS'13	85,06	85,87	1,89
A, C, T'13	91,11	79,10	0,00
A, C, T'14	90,37	74,74	1,14
A, C, T'14s	98,78	64,86	0,00
B, C, LJ'14	65,11	80,59	67,64
B, C, SMS'13	48,98	55,08	88,73
B, C, T'13	65,65	65,39	77,99
B, C, T'14	65,89	62,87	71,00
B, C, T'14s	78,79	32,50	61,54

Table 3: `senti.ue` precision in best mode

task, mode, data	Positive	Negative	Neutral
A, C, LJ'14	80,11	74,70	0,00
A, U, SMS'13	80,62	80,48	11,54
A, C, T'13	85,05	81,16	0,00
A, C, T'14	89,09	68,25	14,29
A, C, T'14s	83,51	88,89	0,00
B, C, LJ'14	77,65	64,99	68,30
B, C, SMS'13	83,68	58,81	74,58
B, C, T'13	78,72	60,93	68,87
B, C, T'14	80,07	49,42	60,28
B, C, T'14s	55,32	76,47	36,36

Table 4: `senti.ue` recall in best mode

5 Conclusions

Continuing last year experience, we participated in SemEval-2014 Task 9 to test our approach for a real-time SA system for the English used nowadays in social media content. We changed the method for subtask A, now considering also the text around the area to classify, by dedicating new features to it, which led to good results. Our method for overall sentiment is ML based, using a restricted set of features that are dedicated to superficial text properties, negation presence, and sentiment lexicons. Without a deep linguistic analysis, our system achieved a reasonable result in subtask B. The evaluation of our solution, in both subtasks, shows an appreciable improvement, by 10% or more, when compared to our results in 2013. We believe that the additional training instances used in unconstrained mode and subtask B, about laptops and restaurants, have a writing style different from most of the test set documents. And perhaps this is the cause for lower score in the unconstrained mode, something that happened also with many systems in the past edition (Wilson et al., 2013).

This time, we implemented the contextual polarity solution based on the subtask B classifier. Given the results, we intend to do, in the near future, a new iteration of our system where the overall classifier will depend on (or receive features from) the current subtask A classifier.

It seems to us that `senti.ue` feature engineering can be improved, maintaining this line of development. Once stabilized, the introduction of named entity recognition and a richer linguistic analysis will help to identify the sentiment target entities, as the ultimate goal for this system.

References

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.

Lee Becker, George Erhart, David Skiba and Valentine Matula. 2013. AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). Atlanta, Georgia, USA.

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

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classication. Journal of Machine Learning Research, 9:1871–1874.

Tobias Günther and Lenz Furrer. 2013. GU-MLT-LT: Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). Atlanta, Georgia, 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.

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013). Atlanta, Georgia, USA.

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.

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

Sara Rosenthal, Alan Ritter, Veselin Stoyanov, and Preslav Nakov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In Proceedings of the Eighth International Workshop on Semantic Evaluation (SemEval'14). August 23-24, 2014, Dublin, Ireland.

José Saias. 2013. In search of reputation assessment: Experiences with polarity classification in replab 2013. In Pamela Forner, Roberto Navigli, and Dan Tufis, editors, *CLEF 2013 Evaluation Labs and Workshop Online Working Notes - Online Reputation Management (RepLab)*, Valencia, Spain, September 2013. ISBN 978-88-904810-5-5.

José Saias and Hilário Fernandes. 2013. senti.ue-en: an approach for informally written short texts in semeval-2013 sentiment analysis task. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 508–512, Atlanta, Georgia, USA.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. Language Resources and Evaluation, 39:165–210.

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*. ACL.