In search of reputation assessment: experiences with polarity classification in RepLab 2013

José Saias

Departamento de Informática, ECT
Universidade de Évora, Portugal
jsaias@uevora.pt

Abstract.
The `diue` system uses a supervised Machine Learning approach for the polarity classification subtask of RepLab. We used the Python NLTK for preprocessing, including file parsing, text analysis and feature extraction. Our best solution is a mixed strategy, combining bag-of-words with a limited set of features based on sentiment lexicons and superficial text analysis.
This system begins by applying tokenization and lemmatization. Then each tweet content is analyzed and 18 features are obtained, related to presence of polarized term, negation before polarized expression and entity reference.
For the first run, the learning and classification were performed with the Decision Tree algorithm, from the NLTK framework. In the second run, we used a pipeline of classifiers. The first classifier applies Naive Bayes in a bag-of-words feature model, with the 1500 most frequent words in the training set. The second classifier used the features from the first run plus another feature with the result from the previous classifier. Our system's best result had 0.54694 Accuracy and 0.31506 in F measure.

1 Introduction

This article describes the participation of a group from the Department of Computer Science, at the University of Évora, in the RepLab track of the 2013 edition of Cross Language Evaluation Forum (CLEF). RepLab[1] is a competitive evaluation exercise for online reputation management systems, organized as a CLEF lab activity[2]. In this challenge, the reputation processing subtasks are:

1. tweet filtering: distinguish the tweets that are related to the entity from those who are not;
2. reputation polarity classification: detect if a tweet has a positive, negative or neutral impact on the entity reputation;

[1] http://www.limosine-project.eu/events/replab2013
[2] http://clef2013.org/index.php?page=Pages/labs.html

3. tweet clustering per entity related topic;
4. priority detection.

Systems can participate in the full monitoring task, with the combined results of the four subtasks, or present partial solutions to the global task, providing results for one or more subtasks.
In this first participation, we focused our attention on the polarity classification subtask, because this seems to be a key task in reputation analysis. We have a recent work in the area of sentiment analysis in social media [1]. Polarity for reputation is different from standard sentiment analysis for two reasons. Firstly, an objective text, without sentiment, may still affect an entity's reputation. And on the other hand, sometimes the polarity of the expressed sentiment may be contrary to the resulting polarity for the reputation of the target entity. Given these differences, we have designed the `diue` system, with a supervised Machine Learning approach for classifying the reputation polarity, as described in section 3. The following section presents some recent related work.

2 Related Work

In the previous edition of RepLab about 10 systems participated in subtask polarity classification for reputation. Most systems rely on a sentiment polarity based approach, adapted for the reputation task [2].
DAEDALUS system [3] has a model with rules and annotated resources for sentiment analysis. It applies an aggregation algorithm to calculate the polarity value based on the individual text segments polarity values. Morphosyntactic analysis is performed, for lemmatize, divide the text and detect negation. The approach from FBM/Yahoo! system [4] relies on lexicon-based techniques and Support Vector Machines classifiers. The UNED system [5] adapts an existing emotional concept-based system for sentiment analysis to determine polarity for entity reputation. Its approach includes the detection of negation and intensifiers, in order to deal with the effect of subordinate sentences.
The ILPS system [6] classifies the polarity of a tweet based on the observation of the reactions to that tweet, such as replies and retweets.

3 Our Experiments

The reputation processing is done on data from Twitter, in English or Spanish. Systems received a corpus of tweets in both languages, arranged in sets for each of the 61 entities [7]. Due to the Twitter's terms of service, the provided corpus did not include the content of tweets, but only the identifier codes, for each system then make its own reading.
Obtaining the tweets was a setback in our participation. The normal download

API imposes a maximum number of hits per hour, being very time consuming. Because of our naivete, we did not anticipate the difficulties of fetching all tweets, and when we completed the process, we had only 24 hours to the end of the official submission period. This left little room for studying the data.
For each entity, it was given its name, the domain which the entity belongs to, and URL addresses of their homepage and Wikipedia entries, in English and in Spanish. Our system did not use the contents of the homepages nor Wikipedia. Additional background tweets for each entity, and external links mentioned in the tweets were also provided for the participating systems, but we lacked the time to prepare that preprocessing step.
The `diue` system uses a supervised Machine Learning approach for the polarity classification subtask of RepLab. As mentioned in section 1, we developed a recent work [1] on Sentiment Analysis in Twitter. Despite differences in polarity for reputation, the data structure and some initial treatment to apply to tweet text are identical. So we decided to use part of the previous procedure, adding features related to the entity reference and its reputation implication.
For the initial entity file handling and parsing, for the text analysis and feature extraction, and also to manage the output format, we used Python and the Natural Language Toolkit (NLTK), a framework with resources and programming libraries suitable for linguistic processing [8, 9].
Tweet text processing started with tokenization, in which the splitting was white space or punctuation based. Lemmatization was then applied through the NLTK WordNet Lemmatizer. Here began the differences in relation to language. This lemmatization would help only tweets in English, because it was not applied similar functionality for Spanish.
To help determine the polarity direction in some terms of the text, our system uses three sentiment lexicons for English terms, and another hand-built resource with 100 words in Spanish. AFINN [10] is a sentiment lexicon containing English words manually labeled by Finn Årup Nielsen, from 2009 to 2011. Words were rated between minus five (negative) and plus five (positive). SentiWordNet [11] is a lexical resource for opinion mining that assigns sentiment scores to each synset of WordNet[3]. We apply a threshold, disregarding terms whose score absolute value is less than 0.3. By doing this, we look for sharper polarities, or greater confidence in the direction of polarity. The third English sentiment lexicon derived from Bing Liu's work [12] on online customer reviews of products.
After tokenization and lemmatization, each tweet content is analyzed for extracting the features to use in machine learning. In the first run, we decided not to use a bag-of-words model. Instead, we chose a more restricted set of 18 features involving:

- presence of polarized term, using sentiment lexicons;
- negation before polarized expression;
- polarized term before entity reference;
- polarized term after entity reference;
- negation before entity reference;

[3] http://wordnet.princeton.edu/

– entity reference followed by negation and polarized term.

Each of the above represents a group of features. The presence of polarized term is checked for all sentiment lexicons, generating a pair of boolean features for each, to signal the presence of an expression with negative polarity and the presence of a positive expression. The system also creates an overall sentiment value feature, determined by consulting all those lexicons and adding 1 or -1, for each polarized term in the tweet, according to the term polarity. The features involving the entity reference try to capture differences that the learning algorithm can then associate to positive or negative impact on reputation.
The learning and classification were performed with the Decision Tree algorithm, from the NLTK framework. Each tweet in the training set is annotated with RELATED/UNRELATED (the tweet is/is not about the entity, for the filtering subtask) and POSITIVE/NEUTRAL/NEGATIVE to train the polarity classification. When training our model, the system discards tweets not having the RELATED annotation, because these have no interest for the subtask.
In preliminary experiments, the accuracy returned by NLTK matched with the result obtained with the evaluation script provided by the organization for use in development phase. This accuracy was around 58%, so we generated the first run over the test data.
In the second run, we used a pipeline of classifiers. The first classifier applies Naive Bayes in a bag-of-words feature model, with the 1500 most frequent words in the training set. The second classifier used the features from the first run, plus one more feature with the result from the former classifier. In this second run, some errors were also corrected in the extraction of features. This was the case of the overall sentiment value calculation, which needed sometimes to invert the polarity of the values, when the source expression was affected by the negation. A small lemmatization related bug was also fixed.
For the last run, a few terms were introduced in the Spanish sentiment lexicon, and the overall sentiment value feature was turned off in the first classifier features.
At the end of the competition, the systems were given extra time to finish ongoing experiments, and also receive the assessment on those latter unofficial runs. Our second and third runs were submitted during this extra period.
Given the short time and the delays in downloading the tweets, our system still got 0.995 for the ratio of tweets in the goldstandard that have been processed. Next section describes the evaluation metrics and the results for the submitted runs.

4 Results

The systems involved in the polarity for reputation classification task are evaluated according to Accuracy, Reliability and Sensitivity. These latter two measures have already been used in RepLab 2012, and are described in [13]. Table 1 has the result of evaluating the three runs for the `diue` system. In the second

column we can see the Accuracy, as the proportion of cases where the system guesses the right polarity class. The F column shows the balanced F measure combining Reliability and Sensitivity. The value shown in these four columns is the average for all entities. Pearson correlation, in the last column, is calculated between average polarity of entities according to the system versus the gold standard.
The last two runs are marked with * because they were submitted in the extra period, and thus were not considered as official runs in competition, despite being assessed.
The Accuracy is practically the same in all three cases, but better in the second run, with 0.54694. Reliability is higher in the first run, with a little difference. Sensitivity, F and Pearson correlation make clear the difference between the first run and the other two using the classifier pipeline, all having the best result in run 2.

Run	Accuracy	Reliability	Sensitivity	F	Pearson correlation
1	0.54688	0.33303	0.21516	0.25467	0.21398
2*	0.54694	0.32923	0.31620	0.31506	0.64769
3*	0.54603	0.32734	0.31470	0.31343	0.64572

Table 1. Evaluation of polarity subtask results for `diue` system

5 Discussion

If we looked only to the Accuracy values, we would eventually say that the runs have equal results, with 54% accuracy. But the results of each run are substantially different, in particular from the first to the other two runs. In the first run the system assigned the neutral polarity to 9804 tweets, while in the second run that number rose to 18586. Run 1 had about 13000 more positive tweets than run 2 and 3.
The pipelined classifier brought the bag-of-words model to complement the previous model and to compensate some scarcity in that feature set. This is noticed in the F and Pearson correlation values evolution.
Let us now compare our modest results with the best systems in competition [7] in the same subtask. The best value of our system accuracy is 0.54694, while the best system managed 0.68596 (and it seems to have had no problems downloading the tweets, having 100% processed tweets). Considering the F measure, the best official result was 0.38166, and the average was 0.22672. Our system official run had 0.25467, and for the second run we got 0.31506.
At first, we thought that the existence of two languages would be a bigger problem. Writing used in tweets is very informal and full of typos. In Spanish tweets can also appear emoticons and may even arise expressions in English that are

commonly used. Therefore certain results could be achieved, even with the base system.

6 Conclusions

This was our first experience in RepLab challenge. Our system is not yet ready to the full reputation monitoring task. We have dedicated our efforts to the polarity classification subtask. Our best solution is a mixed strategy, combining bag-of-words with a limited set of features based on sentiment lexicons and superficial analysis of text.
If we repeated the process, we would have started downloading the tweets earlier, in order to have the time for experiences and analysis, and to choose the most appropriate feature set for this kind of data and purpose.
For future work, we highlight the importance of strengthening the resources of language support for Spanish, including lemmatization, and sentiment lexicon. In the bag-of-words, we used only the 1500 most frequent words in the training set. Maybe we should increase the number of words/features.
We consider NLTK very effective in text processing. For the future, however, we consider using another tool for machine learning, supporting more classification algorithms in the same friendly way, but, at the same time, allowing a greater degree of configuration.
Regardless of the results obtained by our system, we consider that the participation in this challenge was very positive, by its competitive spirit, the large-scale evaluation, and the sharing of new ideas in the treatment of reputation.

References

1. José Saias and Hilário Fernandes. 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, June 2013. Association for Computational Linguistics.
2. Enrique Amigó, Adolfo Corujo, Julio Gonzalo, Edgar Meij, and Maarten de Rijke. Overview of RepLab 2012: Evaluating online reputation management systems. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.
3. Julio Villena-Román, Sara Lana-Serrano, Cristina Moreno, Janine García-Morera, and José Carlos González Cristóbal. Daedalus at replab 2012: Polarity classification and filtering on twitter data. In Forner et al. [14].
4. Jose M. Chenlo, Jordi Atserias, Carlos Rodriguez, and Roi Blanco. Fbm-yahoo! at replab 2012. In Forner et al. [14].
5. Jorge Carrillo de Albornoz, Irina Chugur, and Enrique Amigó. Using an emotion-based model and sentiment analysis techniques to classify polarity for reputation. In Forner et al. [14].

6. Maria-Hendrike Peetz, Maarten de Rijke, and Anne Schuth. From sentiment to reputation. In Forner et al. [14].
7. Enrique Amigó, Jorge Carrillo de Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Tamara Martín, Edgar Meij, Maarten de Rijke, and Damiano Spina. Overview of replab 2013: Evaluating online reputation monitoring systems. In *Fourth International Conference of the CLEF initiative - CLEF 2013 Proceedings, Valencia, Spain*, Springer LNCS, Sep 2013.
8. Edward Loper and Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1*, ETMTNLP '02, pages 63–70, USA, 2002. Association for Computational Linguistics.
9. Jacob Perkins. *Python Text Processing with NLTK 2.0 Cookbook*. Packt Publishing, 2010.
10. Finn Årup Nielsen. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *1st Workshop on Making Sense of Microposts (#MSM2011)*, pages 93–98, 2011.
11. Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).
12. Bing Liu. Opinion observer: Analyzing and comparing opinions on the web. In *In WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM Press, 2005.
13. Enrique Amigó, Julio Gonzalo, and Felisa Verdejo. A general evaluation measure for document organization tasks. In *Proceedings SIGIR 2013*, July 2013.
14. Pamela Forner, Jussi Karlgren, and Christa Womser-Hacker, editors. *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, 2012.