GTI at SemEval-2016 Task 4: Training a Naive Bayes Classifier using Features of an Unsupervised System

Jonathan Juncal-Martínez, Tamara Álvarez-López, Milagros Fernández-Gavilanes
Enrique Costa-Montenegro, Francisco Javier González-Castaño
GTI Research Group
AtlantTIC Centre, School of Telecommunication Engineering, University of Vigo
36310 Vigo, Spain
{jonijm, talvarez, milagros.fernandez, kike}@gti.uvigo.es,
javier@det.uvigo.es

Abstract

This paper presents the approach of the GTI Research Group to SemEval-2016 task 4 on Sentiment Analysis in Twitter, or more specifically, subtasks A (Message Polarity Classification), B (Tweet classification according to a two-point scale) and D (Tweet quantification according to a two-point scale). We followed a supervised approach based on the extraction of features by a dependency parsing-based approach using a sentiment lexicon and *Natural Language Processing* techniques.

1 Introduction

In recent years, research on the field of *Sentiment Analysis* (SA) has increased considerably, due to the growth of user content generated in social networks, blogs and other platforms on the Internet. These are considered valuable information for companies, which seek to know or even predict the acceptance of their products, to design their marketing campaigns more efficiently. One of these sources of information is Twitter, where users can write about any topic, using colloquial and compact language. As a consecuence, SA in Twitter is specially challenging, as opinions are expressed in one or two short sentences.

Many approaches have been proposed for SA, and can be roughly divided into two categories. The first one tries to capture and model linguistic knowledge through the use of dictionaries (Taboada et al., 2011) containing words that are tagged with their semantic orientation. These methods detect the words present in a text using different strategies involving lexics, syntax or semantics (Quinn et al., 2010). The other one is machine learning-based, which is currently the most predominant approach including supervised learning and deep learning. They widely use classifiers including *Support Vector Machines* (SVM), *Maximum Entropy Models* (MAXENT), and *Naive Bayes* classifiers. Most of the time, they are built from features of a "*bag of words*" representation (Pak and Paroubek, 2010).

Our group has participated in SemEval-2016 task 4 on Sentiment Analysis in Twitter, subtasks A (Message Polarity Classification), B (Tweet classification according to a two-point scale) and D (Tweet quantification according to a two-point scale) (Nakov et al., 2016b).

The remainder of this article is structured as follows: Section 2 presents in detail the system proposed for the performance of these subtasks, and Section 3 shows the results obtained and discusses them. Finally, Section 4 summarizes the main findings and conclusions.

2 System Overview

Our main objective was to create a supervised system using extracted features from an unsupervised system described in (Fernández-Gavilanes et al., 2015). This last approach comprises different processing stages, including the generation of sentiment lexicons, test preprocessing and the application of different methods for determining contextual polarity based on syntactical structure. This makes our approach robust in diverse contexts without the need for previous manual tagging of datasets. As we can decide independently which modules

Proceedings of SemEval-2016, pages 115–119,
San Diego, California, June 16-17, 2016.

of the unsupervised system to use or not, it was easy to extract different features from each one individually or together. Once extracted, classification was applied using *Weka* tool (Hall et al., 2009). This environment contains a collection of *machine learning-based* algorithms for data mining tasks, such as, classification, regression, clustering, association rules, and visualization. The new supervised system was built with a *Naive Bayes* classifier.

2.1 Modules combination features

The first extracted features of the unsupervised system were the different sentiment outputs of the modules combination. As mentioned before, modules can be enabled and disabled independently. With this feature, multiple sentiment outputs were obtained from these combinations.

The unsupervised system has four different modules ("*intensification treatment*" (I), "*negation treatment*" (N), "*polarity conflict treatment*" (C) and "*adversative/concessive clause treatment*" (A/CO)). In total, there were 14 possible combinations: one by one, combining pairs or groups of three of them, and all of them at once (the latter is the default output of the unsupervised system). In subtask A, each output obtained is defined by a sentiment value contained between three possible ones: *negative*, *neutral* or *positive*. However, in subtask B, the sentiment value obtained for each combination only can be contained between two possible ones: *negative* or *positive*. So, the result of each one of these 14 combinations was considered as a feature. All of them are defined in Table 1.

Combination	Subtask A	Subtask B
I	POSITIVE NEGATIVE NEUTRAL	POSITIVE NEGATIVE
N		
C		
A/CO		
I + N		
I + C		
I + A/CO		
N + C		
N + A/CO		
C + A/CO		
I + N + C		
I + N + A/CO		
N + C + A/CO		
ALL		

Table 1: Possible combinations of modules.

2.2 Individual modules features

In addition to the previous modules combination results extracted, other features were also extracted from each module independently. Each tweet was represented as a vector of generic and relational features. Generic features are those that are not related to a scope in a given tweet, and relational features represent the corresponding scope needed for each module. For example, in the negation module, the scope would begin in the unigram that caused the negation (the negator term itself), and would cover all affected unigrams in a branch of the dependencies tree, detected by its syntactic function. For this reason, both types of features can be distinguished. The option chosen to mark the scope was to use relational attributes. With them, unigram to unigram can be stored with all its associated features: such as it is an intensifier, a negator, a part of the scope of negation, etc.

Generic features: The first features introduced are not related to a scope, and involve:

- *Phrases*: the number of phrases of a particular tweet.
- *Adjectives*: the number of existing adjectives in a given tweet.
- *Common names*: the number of existing common names in a given tweet.
- *Verbs*: the number of existing verbs in a given tweet (except auxiliary verbs).
- *Positive/negative polarity unigrams*: the number of unigrams with positive/negative polarity in a given tweet.
- *Positive/negative emoticons*: the number of positive/negative emoticons (with positive/negative polarity) in a given tweet.
- *Positive/negative intensifications*: the number of positive/negative intensifications in a given tweet.
- *Unigrams*: all lemmas were considered (except *hashtag*, *mention*, *URL*, unigrams with numbers, unigrams with length 1 and punctuation marks).

Relational features: They can be defined as an array of features. Each unigram of a given tweet has assigned all the features defined in the relational, so it is easy to mark the scope of treatment of each of the separate modules. Then, all features introduced for each unigram in the relational are detailed.

- *Part of speech*: it can take one of the next five values: adjective, common name, verb, adverb or other.
- *Polarity value*: it can take one of the next seven values: negative +, negative, negative -, none, positive -, positive and positive +.
- *Is intensifier*: it indicates if an unigram is an intensifier. It can take one of the next five values: intensity - -, intensity -, none, intensity +, intensity + +.
- *Was intensified*: it indicates if an unigram was intensified. It can take one of the next seven values: negative +, negative, negative -, none, positive -, positive and positive +.
- *Conflict unigram*: it indicates if an unigram causes a polarity conflict, with its polarity converted to intensity. It can take one of the next five values: intensity - -, intensity -, none, intensity +, intensity + +.
- *Affected unigram*: it indicates when an unigram is affected by a conflict unigram, modifying its polarity value. It can take one of the next seven values: negative +, negative, negative -, none, positive -, positive and positive +.
- *Negator unigram*: it indicates when an unigram is a negator, modifying the polarity value of the subsequent unigrams. It can take one of the next two values: 0 if it isn't a negator or 1 if it is.
- *Negated unigram*: it indicates when an unigram is affected by a negator, modifying its polarity value. It can take one of the next seven values: negative +, negative, negative -, none, positive -, positive and positive +. This is the value contributed by that unigram in a negated branch of the dependencies tree (the scope).

2.3 Sentiment prediction

Once features were extracted, the next step was to create a model to predict sentiment in testing datasets. Previously, it was said that *Weka* contains a collection of machine learning algorithms for data mining tasks. Several algorithms were tested, such as *Support Vector Machines* (SVM) (Mullen and Collier, 2004), *Large-Scale Linear* (LIBLINEAR) (Fan et al., 2008) or *Hidden Markov Model* (HMM) (Soni and Sharaff, 2015), but the best results obtained were with *Naive Bayes* (Tan et al., 2009). Also, *10-fold cross-validation* was used to obtain the best classification model with the training dataset. Once all classification models were obtained, in subtask A the model with the best *F-measure* was selected, while in subtask B the selected model was the one with the best *recall* (R), as the organization proposed. For the subtask D, the subtask B results were taken into account.

A previous step before the selection of the best classification model is needed. Most algorithms do not accept as input relational attributes, so it was necessary to apply an unsupervised filter by attribute, *RELAGG*, both in training and test files. It processes all relational attributes that fall into the user defined range, making them nominal attributes. In *Naive Bayes* algorithm, the default settings were used, for both the training and the testing datasets, as they are defined in *Weka*. Finally, applying the best model for each subtask on the corresponding testing dataset, the final sentiment prediction for all tweets was obtained.

3 Experimental results

In this section, the conducted experiments for subtasks A, B and D are described. The experiments were carried out using the datasets provided by SemEval-2016 task organizers. These datasets are composed of texts extracted from Twitter, and in the case of the subtasks B and D, with a given topic. In subtask A, the number of tweets is `32009` and the performance of the system is measured by means of the *F-score*. In subtask B, the number of tweets is `10551` and the performance of the system is measured by means of the *macroaveraged recall*. Finally, in subtask D, as in subtask B, the number of tweets is `10551` (same dataset) but this time, the

performance of the system is measured by means of the normalized cross-entropy, better known as *Kullback-Leibler Divergence* (KLD). In this last case, there is a minor modification in the formula, with a smoothed version of the originals $p(c_j)$ and $\hat{p}(c_j)$, and a smoothing factor ϵ. All of these measurements are described in (Nakov et al., 2016a).

Table 2 presents the overall scores for subtasks A, B and D, in their respective test sets: *F-measure*, *recall* and KLD, respectively. The third column shows the unsupervised approach results (UAR) and the fourth shows the supervised approach results (SAR) obtained this year.

	Test set	UAR	SAR
Subtask A	*Tw 2013*	59.44%	61.17%
	SMS 2013	52.19%	52.38%
	Tw 2014	62.45%	63.90%
	TwS 2014	46.77%	46.77%
	LJ 2014	61.16%	62.32%
	Tw 2015	57.64%	58.44%
	Tw 2016	53.17%	53.94%
Subtask B	*Tw 2016*	73.36%	73.60%

	Test set	UAR	SAR
Subtask D	*Tw 2016*	0.067	0.055

Table 2: Results of the approach for subtasks A, B and D. *Tw* refers to Twitter, *TwS* to Twitter Sarcasm and *LJ* to LiveJournal.

After performing several experiments on the training, development and development-test datasets provided by organizers, the neutral sentiment intervals were set to [-0.5, 0.5] for subtask A and [-0.05, 0.05] for subtask B (subtask D depends on subtask B). More specifically, in subtask A, our supervised approach was tested with SemEval-2014 development-test, SemEval-2015 development-test and 2016 development-test datasets provided; in subtask B, it was tested with 2016 development-test dataset; and for subtask D, the 2016 development-test dataset results in subtask B were taken into account. In development time, the improvement of our supervised system was between 1 and 3 % compared to our unsupervised system for subtasks A and B, and for subtask D a difference of -0.02 KLD.

In order to assess the improvement of our supervised system regarding our unsupervised system, a comparison is performed in the test sets of this year, as it can be seen in Table 2. With these results, we can say that the new approach, in most cases, improves the unsupervised system, between 0.19 and 1.73 % for subtask A and B (except in Twitter Sarcasm 2014), and a difference of -0.012 in subtask D.

4 Conclusion

This paper describes the participation of the GTI Research Group, AtlantTIC Centre, University of Vigo, in SemEval-2016 task 4: Sentiment Analysis in Twitter. The results were achieved using a supervised system with extracted features from an unsupervised system, described in (Fernández-Gavilanes et al., 2015). Table 3 shows the position of this approach in the ranking published for subtasks A, B and D for the datasets evaluated.

	Test set	Position
Subtask A	*Twitter 2013*	13 / 34
	SMS 2013	17 / 34
	Twitter 2014	16 / 34
	Twitter Sarcasm 2014	7 / 34
	LiveJournal 2014	17 / 34
	Twitter 2015	19 / 34
	Twitter2016	20 / 34
Subtask B	*Twitter 2016*	9 / 19
Subtask D	*Twitter 2016*	5 / 14

Table 3: Positions of the approach for subtasks A, B and D.

The unsupervised approach consists of sentiment propagation rules on dependencies where features were selected (as the different sentiment outputs of the modules combination), and a vector of generic (features not related to a scope in a given tweet) and relational (features extracted from the scope in each treatment performed in each module) features. The results denote a low/medium improvement in subtask A regarding the unsupervised system, and a low improvement in the subtask B (also reflected in the subtask D). Although the new approach is supervised, the fact of using only features of an unsupervised system makes it totally different from other approaches, and still has margin of improvement adding new external features.

Acknowledgments

This work was supported by the Spanish Government, co-financed by the European Regional Development Fund (ERDF) under project TACTICA.

References

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.*, 9:1871–1874, June.

Milagros Fernández-Gavilanes, Tamara Álvarez López, Jonathan Juncal-Martínez, Enrique Costa-Montenegro, and Francisco Javier González-Castaño. 2015. GTI: An Unsupervised Approach for Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 533–538, Denver, Colorado, June. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Tony Mullen and Nigel Collier. 2004. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 412–418, Barcelona, Spain, July. Association for Computational Linguistics.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016a. Evaluation Measures for the Semeval-2016 task 4: Sentiment Analysis in Twitter (Draft: Version 1.12). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016b. SemEval-2016 task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus 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. European Language Resources Association (ELRA).

Kevin M. Quinn, Burt L. Monroe, Michael Colaresi, Michael H. Crespin, and Dragomir R. Radev. 2010. How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, 54(1):209–228.

Swati Soni and Aakanksha Sharaff. 2015. Sentiment Analysis of Customer Reviews Based on Hidden Markov Model. In *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*, ICARCSET '15, pages 12:1–12:5, New York, NY, USA. ACM.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, June.

Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. 2009. Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 337–349, Berlin, Heidelberg. Springer-Verlag.