DsUniPi: An SVM-based Approach for Sentiment Analysis of Figurative Language on Twitter

Maria Karanasou
Dept. of Digital Systems
University of Piraeus
Greece
karanasou@gmail.com

Christos Doulkeridis
Dept. of Digital Systems
University of Piraeus
Greece
cdoulk@unipi.gr

Maria Halkidi
Dept. of Digital Systems
University of Piraeus
Greece
mhalk@unipi.gr

Abstract

The DsUniPi team participated in the SemEval 2015 Task#11: Sentiment Analysis of Figurative Language in Twitter. The proposed approach employs syntactical and morphological features, which indicate sentiment polarity in both figurative and non-figurative tweets. These features were combined with others that indicate presence of figurative language in order to predict a fine-grained sentiment score. The method is supervised and makes use of structured knowledge resources, such as SentiWordNet sentiment lexicon for assigning sentiment score to words and WordNet for calculating word similarity. We have experimented with different classification algorithms (Naïve Bayes, Decision trees, and SVM), and the best results were achieved by an SVM classifier with linear kernel.

1 Introduction

Sentiment analysis on figurative speech is a challenging task that becomes even more difficult on short social-media related text. Tweet text can be rich in irony that is either stated with hashtags explicitly (such as #irony) or implied. Identifying the underlying sentiment of such text is challenging due to its restricted size and features such as use of abbreviations and slang. Consequently, assigning positive or negative polarity is quite a difficult task. The actual meaning can be very different than what is stated, since, for example, in ironic language what is said can be the opposite of what it is meant. To address this challenge, we propose a system for sentiment analysis of figurative language, which relies on feature selection and trains a classifier to predict the label of a tweet. Given a labelled trial set, the objective of the system is to correctly determine how positive, negative or neutral a tweet is considered to be on a scale of [-5, 5].

2 Related Work

Tweets have unique characteristics compared to other text corpora, such as emoticons, abbreviations, and hashtags. Use of emoticons is considered a reasonably effective way to conveying emotion (Derks et al. 2008, Thelwall et al.). Go et al. (2009) show that machine learning algorithms achieve accuracy above 80% when trained with emoticon data. It is also indicated that the use of hashtags and presence of intensifiers, such as capitalization and punctuation, can affect sentiment identification (Kouloumpis et al., 2010). According to Agarwal et al. (2011) such features can add value to a classifier, but only marginally. Additionally, natural language related features, such as part-of-speech tagging and use of lexicon resources, can significantly contribute to detecting the sentiment of a tweet. Moreover, features that combine the prior polarity of words and their parts-of-speech tags are considered most useful.

The problem of sentiment analysis on figurative language has been addressed in many ways. Researchers have investigated the use of lexical and syntactic features in order to identify figurative language and classify the conveyed sentiment. The complexity of such a task is high, especially given the fact that irony and sarcasm are frequently mixed. Sarcasm is usually used for putting down the target of the comment and is somewhat easier to detect. Irony works as a negation, and it can be

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 709–713,
Denver, Colorado, June 4-5, 2015.

conveyed through a positive context, which makes it difficult to understand the actual meaning of a tweet (Reyes et al. 2012, Veale et al. 2010). Davidov et al. (2010) examined hashtags that indicated sarcasm to identify if such labelled tweets can be a reliable source of sarcasm. They concluded that user-labelled sarcastic tweets can be noisy and constitute the hardest form of sarcasm. Riloff et al. (2013) identify sarcasm that arises from the contrast between a positive sentiment referring to a negative situation. Reyes et al. (2012) involved in their work features that make use of contextual imbalance, natural language concepts, syntactical and morphological aspects of a tweet. Many studies exploit the use of contextual imbalance detection through calculation of semantic similarity among the words. This is achieved using lexical resources, such as WordNet or Whisel's dictionary, and the goal is to identify features like emotional content, polarity of words and pleasantness, adverbs implying negation or expressing timing. Shutova et al. (2010) have deployed an unsupervised method to identify metaphor using synonymy information from WordNet. Reyes et al. (2013) argue that other features such as punctuation marks, emoticons, quotes, and capitalized words, n-grams and skip-grams are also useful to the sentiment analysis process. Moreover, patterns such as "As * As *" or "about as * as *" have been shown to be useful in detecting ironic similes (Veale et al. 2010).

3 Approach

The proposed system consists of two main modules: (a) the preprocessing, and (b) the classification module. Each tweet *t* was submitted to preprocessing, in order to remove useless information and extract the desired/targeted features *f*. The result of the preprocessing of a given tweet *t* consists of a *feature dictionary* (*fd)* that stores the values calculated for each feature. In the classification part, the feature dictionaries are converted to vectors and the result matrix is converted to a term-frequency matrix. The aforementioned process is the same for trial and test data and the tf matrices are used by a classifier for training and prediction. We tested different classifiers, including Naïve Bayes, Decision trees, and SVM, in order to study their performance and select the best-performing.

3.1 Preprocessing

Each tweet is given as input to the preprocessing module, in order to transform it to a feature-value dictionary representation:

$$fd_t = \{f1{:}v1,\ \ldots,\ fn{:}vn\} \quad (1)$$

The preprocessing includes cleaning, which starts with the removal of non-ascii characters and is followed by the detection of certain features. Feature detection takes place before the actual cleaning of the text in order to avoid loss of information, such as punctuation, urls and emoticons. This process checks if a tweet contains question marks or exclamation marks, capitalized words, urls, negations, laughing, retweet, emoticons and hashtags. The last two are categorized concerning the sentiment they may convey. We manually categorized the top20 emoticons and some minor variations (http://datagenetics.com/blog/october52012) as positive or negative, whereas hashtags are categorized as positive, negative or neutral. Hashtag categorization makes use of SentiWordNet score (*swnScore*) and the result is a representation of all the hashtags present in a tweet.

In the hashtag categorization process, if a hashtag *ht* is spelled correctly, its swnScore is retrieved. Otherwise, spellchecking (Kelly) is tried once and if it fails then the hashtag is categorized as neutral. The result depends on the number of positive, negative, neutral hashtags in *HTt* as follows:

$$HTEm_t = \begin{cases} HT\_pos, & c(htPos) > c(htNeg) > 0 \\ HT\_neu, & c(htPos) = c(htNeg) = 0 \\ HT\_neg, & c(htNeg) \geq c(htPos) > 0 \end{cases} \quad (2)$$

where *c(htPos)*, *c(htNeg)* denote the count of positive and negative hashtags in a tweet *t* respectively.

Motivated by the "*As * as **" pattern and after studying the data set, we further identify in the feature selection process the presence of patterns such as "*Don't you**", "*Oh so*?*" and "*As * As **". Cleaning proceeds with punctuation, stop-words, urls, emoticons, hashtags and references removal. Additionally, multiple consecutive letters in a word are reduced to two. Finally, spellchecking is performed to words that have been identified as misspelled in order to deduce the correct word. After cleaning, the process continues with part of speech

(POS) tagging. POS-tagging is performed with the use of a custom model (Derczynski et al., 2013) and simplified tags (NN, VB, ADJ, RB). Words that belong to the same part of speech are used in semantic text similarity calculation sim_t. For this feature, different similarity measures (Resnik's, Lin's, and Wu & Palmer's) provided by nltk are used (Pedersen et al., 2008). The value sim_t is calculated as the maximum similarity score of every combination of two words and their synonyms.

$$sim_t = \frac{\sum sim_V + \sum sim_N + \sum sim_A + \sum sim_R}{c(V)+c(N)+c(A)+c(R)} \quad (3)$$

$$sim_A = \begin{bmatrix} max(sim(A_i, A_{i+1})), & \dots \\ max(sim(A_{n-1}, A_n)) \end{bmatrix} \quad (4)$$

where V, N, A, and R denote the sets that contain the total words that have been identified as verbs, nouns, adjectives and adverbs respectively, while $max(sim(A_i, A_{i+1}))$ is the maximum similarity between the processed words and their *n* synonyms.

Finally, the SentiWordNet score for each word in a tweet is calculated (Baccianella et al., 2010), ignoring words that have fewer than two letters. If the score of a word cannot be determined, then we calculate the SentiWordNet score of the stemmed word. Given that the word wi occurs j times in the SentiWordNet corpus, the total score of wi is given by

$$swnScore_{wi} = \frac{\sum_{k=1}^{j} 1 + wScore(i,k)_p - wScore(i,k)_n}{j} \quad (5)$$

where $wScore(i, k)_p$ and $wScore(i, k)_n$ is the k-th positive (PosScore) and negative (NegScore) score respectively of wi in SentiWordNet. The index i of each word was used in an attempt to correlate each word's position with the calculated sentiment.

Moreover, the total score of a tweet t is calculated as the average of SentiWordNet scores of the words in t.

The result is a dictionary with feature names as keys and values that indicate feature existence. Table 1 depicts the set of features considered by our system, together with the domain of values that they take.

3.2 Classification

For the classification process, the feature dictionaries fd_t of each data set were processed by a vectorizer to produce a vector array (http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.DictVectorizer.html). From the vector array, a term-frequency matrix is calculated (with the use of a TfidfTransformer and the parameter "use_idf" set to False: http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html) and is given as input for training to the chosen classifier. This frequency matrix is used to make predictions about the test set.

Feature	Value
Oh so* (*)	True/ False
Don't you*(*)	True/ False
As*As*(*)	True/ False
Question mark(*)	True/ False
Exclamation - mark(*)	True/ False
Capitals(*)	True/ False
Reference(*)	True/ False
RT	True/ False
Negations(*)	True/ False
URL	True/ False
HT_pos(*)	True/ False
HT_neg(*)	True/ False
HT_neu(*)	True/ False
Emoticon Pos(*)	True/ False
Emoticon Neg(*)	True/ False
POS-tags(*)	"NN", "VB", "ADJ","RB"
$swnScore_{wi}$(*)	"positive", "somewhat positive", "neutral", "negative" "somewhat negative"
swnScoreTotal	"positive", "somewhat positive", "neutral", "negative" "somewhat negative"
sim_t (Resnik*)	Decimal score

Table 1: Calculated features with their value.

4 Experiments and Results

The SemEval data set consists of 9000 tweets that are rich in figurative language and stemmed from

user-generated tags, such as “#sarcasm" and “#irony". There is a 90-10 split for trial and test data. We retrieved 8529 tweets in total, 7606 from the trial set and 923 from the test set. Out of these data sets, positive tweets in total are 8,2%, negative tweets are 85,2% and neutral 6,6%.

4.1 Experiments

We experimented by incrementally adding features, and trying different classifiers. The results of the features that seem to contribute most were used to make the prediction with which the system participated in the task and are the ones marked with (*) in Table 1. It is also worthwhile mentioning that, after trials, discretization was applied to $swnScore_{wi}$ as follows:

$$swnScore_{wi} = \begin{cases} positive, & (> 1.2) \\ somewhat\ positive, & (> 0.05 \leq 1.2) \\ neutral, & (\leq 0.05 \geq 0.95) \\ somewhat\ negative, & (< 0.95 \geq 0.2) \\ negative, & (< 0.2) \end{cases} \quad (6)$$

4.2 Final Results

We evaluate the performance of our approach measuring the cosine similarity between the output of our system and the given scores for the test data set. Other measures such as accuracy, precision and recall are also used in our study.

The most useful features are pos-tags and SentiWordNet score. Semantic similarity (Resnik measure) and hashtags also seem to contribute and the rest of the selected features contribute marginally. These results are coherent with sentiment analysis literature where prior polarity along with POS-tagging seem to add most value to a classifier, and other features like emoticons add up only marginally (Agarwal et al., 2011, Kouloumpis et al., 2010).

Table 2 shows the evaluation results (cosine similarity and accuracy) of our system for both initial and final data set. We can observe that Linear SVM (default parameters: http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html) achieves the best performance with respect to tweets classification. For the final submission, the total of the test and trial sets were used as input for the learning process of the classifier and only one run was submitted. The analysis of the results of the final submission, presented in Table 3, suggests that predictions on ironic and sarcastic tweets are more accurate than tweets that contain metaphor those that do not contain figurative language.

Classifiers	**Decision Tree**		**Naïve Bayes**		**Linear SVM**	
trials/ final	**t**	**f**	**t**	**f**	**t**	**f**
Cosine	0.68	0.45	0.70	0.55	0.78	0.60
Accuracy	0.31	0.21	0.33	0.23	0.38	0.29

Table 2: The results of the classifiers used on the initial test data set (t) and the final (f), with the selected features of the final submission.

	Cosine Similarity	**MSE**
Overall	0.601	3.925
Sarcasm	**0.87**	**1.499**
Irony	**0.839**	**1.656**
Metaphor	0.359	7.106
Other	0.271	5.744
Rank	10	10

Table 3: The final results by category.

5 Conclusion

The proposed system combines structured knowledge sources along with common tweet and figurative text features. A supervised learning approach is followed, having as goal to classify tweets containing irony and metaphors. The system ranked 10th (out of 15) based on both the cosine similarity measure and MSE. Among ironic, sarcastic, metaphoric and others, the best results were achieved in tweets containing irony and sarcasm. The most useful features for learning are pos-tags, Senti-WordNet score, text semantic similarity and hashtags. Our study shows that the performance of our system could be improved by adding features related to metaphor and considering better use of hashtags in the classification process. Besides, the use of non-figurative tweets in learning can significantly contribute to classify tweets that do not contain figurative language.

Acknowledgements

The work of C. Doulkeridis and M. Halkidi has been co-financed by ESF and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) - Research Funding Program: Aristeia II, Project: ROADRUNNER.

References

Antonio Reyes, Paolo Rosso, Davide Buscaldi (2012). From Humor Recognition to Irony Detection: The Figurative Language of Social Media. Data & Knowledge Engineering 74:1-12.

Yanfen Hao, Tony Veale (2010). An Ironic Fist in a Velvet Glove: Creative Mis-Representation in the Construction of Ironic Similes. Minds and Machines 20(4):635–650.

Antonio Reyes, Paolo Rosso, Tony Veale (2013). A Multidimensional Approach for Detecting Irony in Twitter. Languages Resources and Evaluation 47(1): 239-268.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, Ruihong Huang. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013).

Dmitry Davidov, Oren Tsur, and Ari Rappoport (2010). Semi-supervised recognition of sarcastic sentences in twitter and amazon. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL 2010.

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, Antonio Reyes, John Barnden (2015). SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In: Proc. Int. Workshop on Semantic Evaluation (SemEval-2015), Co-located with NAACL and *SEM, Denver, Colorado, US, June 4-5, 2015.

Ekaterina Shutova, Lin Sun and Anna Korhonen (2010). Metaphor Identification Using Verb and Noun Clustering. In: Proceedings of the 23rd International Conference on Computational Linguistics.

Alec Go, Richa Bhayani, and Lei Huang (2009). Twitter sentiment classification using distant supervision. In: Proceeding LSM '11 Proceedings of the Workshop on Languages in Social Media Pages 30-38.

Daantje Derks, Arjan E. R. Bos, and Jasper von Grumbkow (2007). Emoticons and online message interpretation. Social Science Computer Review, 26(3), 379-388.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, and Di Cai, Arvid Kappas (2010). Sentiment Strength Detection in Short Informal Text. Journal of the American Society for Information Science and Technology Volume 61, Issue 12, pages 2544–2558, December 2010

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore (2011). Twitter sentiment analysis: The Good the Bad and the OMG! In: Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, Proceedings of the Fifth International Conference on Weblogs and Social Media, ICWSM' 11, pages 538–541, Barcelona, Spain.

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau (2011). Sentiment Analysis of Twitter Data. In: LSM'11 Proceedings of the Workshop on Languages in Social Media Pages 30-38.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva (2013). Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing, ACL.

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 7th Conference on Language Resources and Evaluation (LREC 2010), Valletta, MT, 2010, pp. 2200-2204.

Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. (2004). Wordnet::similarity - measuring the relatedness of concepts. In: Demonstration papers at HLT-NAACL, pages 38-42.

Fabian Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. In Journal of Machine Learning Research 12, pp. 2825-2830.

Steven Bird, Ewan Klein, and Edward Loper (2009), Natural Language Processing with Python, O'Reilly Media.

Ryan Kelly, https://pythonhosted.org/pyenchant/, v. 1.6.5.