UCSC-NLP at SemEval-2017 Task 4: Sense n-grams for Sentiment Analysis in Twitter

José Abreu, Iván Castro, Claudia Martínez, Sebastián Oliva
Catholic University of the Most Holy Conception
Alonso de Ribera 2850, Concepción, Chile
{joseabreu,cmartinez}@ucsc.cl
{ilcastro,snoliva}@ing.ucsc.cl

Yoan Gutiérrez
University of Alicante
Carretera de San Vicente s/n
Alicante, Spain
ygutierrez@dlsi.ua.es

Abstract

This paper describes the system submitted to SemEval-2017 Task 4-A Sentiment Analysis in Twitter developed by the UCSC-NLP team. We studied how relationships between sense n-grams and sentiment polarities can contribute to this task, i.e. co-occurrences of WordNet senses in the tweet, and the polarity. Furthermore, we evaluated the effect of discarding a large set of features based on char-grams reported in preceding works. Based on these elements, we developed a SVM system, which exploring SentiWordNet as a polarity lexicon. It achieves an $F_1 = 0.624$ of average. Among 39 submissions to this task, we ranked $10th$.

1 Introduction

To determine whether a text expresses a POSITIVE, NEGATIVE or NEUTRAL opinion has attracted an increasingly attention. In particular, sentiment classification of tweets has immediate applications in areas such as marketing, political, and social analysis (Nakov et al., 2016)

Different approaches have shown to be very promising for polarity classification of tweets such as Convolutional Neural Networks trained with large amounts of data (Deriu et al., 2016).

Several authors have studied Machine Learning approaches based on lexicon, surface and semantic features. The proposal of Mohammad et al. (2013) as well as an improved version of Zhu et al. (2014) show very competitive scores.

The latter approach was re-implemented by Hagen et al. (2015) as a part of an ensemble of twitter polarity classifier which is top-ranked in the SemEval 2015 Task 9: Sentiment Analysis in Twitter. Our system proposes to enrich the set of features used by Mohammad et al. (2013). We describe here only the features more relevant for our experiments, further details in all features could be found in Mohammad et al. (2013); Hagen et al. (2015).

- Lexicon Based Features (LB)

NRC-Emotion, NRC-Sentiment140, NRC-Hashtag (Mohammad et al., 2013), BingLiu (Hu and Liu, 2004) and MPQA (Wilson et al., 2005) lexicons have been used to generate features. Given a tweet, the following features were computed:

- Number of words with positive score
- Number of words with negative score
- Sum of the positive scores
- Sum of the negative scores
- Maximum positive score
- Minimum negative score
- Score of the last positive word
- Score of the last negative word

For unigrams, bigrams and non-contiguous pairs were computed separated feature sets.

- N-gram Based Features (WG and CG)

Each 1 to 4-word n-gram present in the training corpus is associated with a feature which indicates if the tweet includes or not the n-gram. For characters, all different occurrences of 3 to 5 grams are considered.

Given its definition, the number of generated n-gram based features is variable and related with the training corpus. In experiments with SemEval

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 807–811,
Vancouver, Canada, August 3 - 4, 2017.

2017 training data, we got near three million of features of this type that is much largest than the number of tweets.

- Cluster Based Features (CB)

For each one of the 1000 clusters identified by Owoputi et al. (2013) using Brown algorithm (Brown et al., 1992) a feature indicates whether the terms of the tweet belong to them.

Mohammad et al. (2013) studied the effect of removing individual set of features as well a whole group of them. Empirical results suggest that lexicon and n-gram based features are the most important since removing them causes the greatest drop on the classifier efficacy measured as the macro-average F-score in the test set.

In this work, we studied how to reduce the number of generated features by removing some of the n-gram based. Next sections describe further details of our approach.

2 System Description

We trained a Support Vector Machine (SVM) as in (Mohammad et al., 2013; Zhu et al., 2014; Hagen et al., 2015). SVM algorithm has proved to be very effective in the Sentiment Analysis task. Moreover, to better assess the effect of the removal or inclusion of new features we decided to use the same classifier as the aforementioned authors.

In the first stage of our system, the tweets were preprocessed like Hagen et al. (2015). To avoid missing some emoticon symbols we ensure UTF-8 encoding in all stages. In addition, instead of detect emoticons using a regular expression [1] we use the tag provided by the CMU pos-tagging tool. In our case, negation was not considered to generate the word n-gram features.

2.1 New Predictor Features

We aim to explore the relation between the polarity and the presence or not of certain sense combinations in the text. Due to synonymy, two semantically equivalent tweets could lead to very different word n-grams while the sense n-grams could be the same in both tweets.

After a word sense disambiguation (WSD) stage, we generated a new version of the tweet where each word is replaced by its sense. A set of new n-grams features are computed using the new text. This approach allows one sense n-gram to represent two or more different word n-grams if the words have the same sense.

To enrich our model respect to those in (Mohammad et al., 2013; Hagen et al., 2015) we have considered SentiWordNet (Baccianella et al., 2010) as a polarity dictionary, idea explored in (Günther and Furrer, 2013). In this case, after WSD, we can use SentiWordNet to compute positive or negative scores for a given word generating features as with the other lexicons.

Considering that elongated (e.g. greaaaat) words could emphasize the sentiment expressed, similar features were computed but only allowing for the lengthened words in the tweet. In this case, we not considered bi-grams lexicons and normalized the elongated words before query the lexicons.

Finally, we studied the following set of new features.

- Additional Features

  - Sense n-grams (SG): one feature for each sense n-gram in the training corpus.

  - SentiWordNet polarity scores (SW): eight features similar to those defined to other lexicons in section 1.

  - Polarity scores of elongated words (EW): eight features similar to those defined to other lexicons in section 1 but only considering lengthened words if any. All lexicons but NRC-Sentiment140 and NRC-Hashtag for bi-grams were used.

  - Polarity of the last emoticon (LE), if any, accordingly to Hogenboom et al. (2015).

2.2 Model Ensemble

With the available training data, we trained several models using different combinations of feature types. Our final submission was an ensemble of the top 10 models trained. Classifiers was combined by weighted voting as explained by Kuncheva (2004). To classify a tweet, we query a model that output a single label and a weight for that label, proportional to the accuracy of the classifier for that class in previous tests. Querying the 10 models, the final classification of the tweet is the most voted class.

Given A_{ij}^{C} the accuracy of the model i over the class C in test data j the weight of that category for

[1] http://sentiment.christopherpotts.net/tokenizing.html

C is computed as $w_i^C = \frac{\sum_{j=1}^{S} A_{ij}^C}{\sum_{m=1}^{M}\sum_{j=1}^{S} A_{mj}^C}$ where $j = 1$ refers to SemEval 2013 test data and so on to $S = 4$ and $M = 10$ is the number of models in the ensemble.

The next section describes the experiments we carried out to assess different feature sets, how weights were computed as well the results.

3 Experiments

Our predictor is based in an ensemble of Support Vector Machines with linear kernel, and $C = 0.005$ trained with all the features proposed by Mohammad et al. (2013); Hagen et al. (2015) plus the new ones detailed in section 2.1. LibLINEAR (Fan et al., 2008) implementation available in Weka (Frank et al., 2016) was used.

As Mohammad et al. (2013), we want to evaluate how removing n-gram and cluster based features affect the results of our models. Table 1 show eight base models resulting of removing combinations of features of the types WG, CG and CB; with X indicating the characteristic set included in the model.

Table 1: Base models

	1	2	3	4	5	6	7	8
WG	x	x	x	-	x	-	-	-
CG	x	x	-	x	-	x	-	-
CB	x	-	x	x	-	-	x	-

Table 2 show different arrangements of the new features which were combined with the based models for a total of 96 experiments.

Table 2: Combinations of new features.

Exp	1	2	3	4	5	6	7	8	9	10	11	12
SG	-	-	-	x	x	-	x	x	x	x	x	x
SW	-	-	-	-	x	-	-	x	-	x	-	x
EW	-	-	x	-	-	x	-	-	x	x	x	x
LE	-	x	-	-	-	x	x	x	-	-	x	x

We replicated twice the experiments that included SG, one time disambiguating with Lesk (Lesk, 1986) algorithm and other considering the most frequent sense for a word. In all experiments, we used implementations from the NLTK (Bird et al., 2009) to disambiguate. In total, 160 different models were evaluated. Note that some of these models just augmented the features in (Mohammad et al., 2013) with some of the new ones.

With the training data of previous SemEval, 2013 to 2016, we mock our participation in these competitions. We trained SVMs for each model and evaluated it with the corresponding test data using the F_1 score for the POSITIVE class. Table 3 show the best (B) and the worst (W) results for each test dataset.

These results allowed us to rank the models. A final ranking was computed averaging the different positions across different test data of the same model. However, a drawback of this approach is that, besides one model could be ranked better than other, the result difference between them could be very small. The 10 top ranked models are the result of the based model 3 (character n-grams discarded) combined with new features $[4, 2, 5, 9, 4*, 12, 9*, 8, 10, 1]$ where * indicates that the WSD was using Lesk algorithm.

Given the results in all previous SemEval test data, the accuracy over each category was obtained for each model as well the weights for the top 10.

Finally, the system submitted was built as follow. We train versions of each of the top 10 models using the SemEval-2017 training data. After removing duplicates, we get $52,780$ tweets. The 10 trained classifiers were combined by weighted voting with weights computed as explained before. Table 4 show results for each category over the $12,284$ test tweets. As regard of the measures used to evaluate systems, our proposal gets an average recall of $\rho = 0.642$, $F_1^{PN} = 0.624$ and accuracy $Acc = 0.565$. The submitted system stood 10th among participants. Further details about the train and test datasets and results of other participants can be found in (Rosenthal et al., 2017)

4 Conclusions and Future Works

Our proposal is based in (Mohammad et al., 2013). We assessed a new set of features as well analyzed the effect of removing some of the features used in this system.

Data in Table 3 as well the top 10 model trained show that the inclusion of the new features cold improve results.

Experiments in (Mohammad et al., 2013) suggest that removing character n-grams attributes degrades the classifier outcome. We also got these results, but when the feature set is extended with the new ones, character n-grams exclusion seems to be convenient. A look of model results and rankings, show that all models in the top 10, furthermore, in the top 30 are models where character

Table 3: Best (B) and worst (W) results in previous SemEval test data. In parenthesis, the number of the base model. An * indicates that the WSD was using Lesk algorithm.

	2013		2014		2015		2016	
	B	W	B	W	B	W	B	W
1	70.82 (3)	65.34 (8)	70.85 (3)	67.58 (6)	66.02 (3)	61.13 (8)	59.30 (3)	56.80 (8)
2	70.92 (3)	65.24 (8)	70.77 (3)	63.67 (8)	63.37 (3)	58.85 (8)	59.00 (3)	55.30 (8)
3	70.52 (3)	65.58 (8)	70.6 (3)	64.3 (8)	65.66 (3)	61.89 (8)	59.40 (3)	56.70 (8)
4	71.33 (3)	66.75 (8)	71.64 (3)	67.54 (2)	66.24 (3)	61.67 (8)	59.5 (3)	57.10 (8)*
5	70.89 (3)	67.39 (8)	71.87 (7)	67.89 (4)	65.83 (3)	62.02 (8)	59.5 (3)	57.10 (8)
6	70.77 (3)	67.5 (8)	71.97 (7)	67.90 (6)	63.44 (1)	58.46 (8)	59.00 (3)	55.50 (8)
7	70.74 (3)	67.46 (8)	71.77 (7)	68.05 (4)	64.84 (3)	61.37 (8)	59.10 (3)	57.30 (6)
8	70.98 (3)	67.63 (8)	71.79 (7)	68.08 (2)	63.57 (3)*	60.4 (8)*	59.00 (3)*	57.30 (8)*
9	71.26 (3)	67.58 (8)*	71.86 (7)*	68.36 (6)*	65.96 (3)	61.76 (8)	59.50 (3)*	57.20 (8)*
10	70.96 (3)	67.62 (8)	72.01 (7)	68.09 (4)	65.88 (3)*	62.05 (8)*	59.60 (3)*	57.10 (8)
11	70.83 (3)	67.48 (8)	72.05 (7)	67.96 (4)	63.71 (3)*	60.80 (8)*	59.10 (3)	57.50 (6)
12	70.94 (3)	67.68 (8)	71.82 (7)*	68.25 (6)*	63.63 (3)*	60.31 (8)*	59.00 (3)*	57.40 (8)*

Table 4: Results in SemEval 2017 test, Precision (P), Recall (R) and F1.

	P	R	F1
POSITIVE	0.4505	0.8156	0.5804
NEGATIVE	0.5694	0.8072	0.6678
NEUTRAL	0.7617	0.3020	0.4325

n-grams were excluded but some of the new ones considered.

Another interesting fact is that systems seem to be more sensitive to word n-grams and cluster based attributes. The best ranked model without n-grams, stood 23 in our ranking. Character n-grams were also omitted in this model, which was extended with SG, SW and LE features. After the release of the gold labels, we evaluated the predictions of other models not submitted but also trained with the SemEval 2017 training data. The aforementioned model shows a $F_1^P N = 0.652$, better than the model we submitted. It is important to say that this model used only $822,650$ features, substantially less than the $2,993,189$ used by the best of our single models over test data which only discards character n-grams plus SG, EW and LE features and achieves a $F_1^P N = 0.654$

These results open an interesting direction of future work, further study how to minimize the set of features used without a noticeable degradation of prediction results. Ideally, identifying a set of features of size independent of the corpus as the lexicon based ones.

Acknowledgments

This paper has been partially supported by the Catholic University of the Most Holy Conception through the research project DIN-01/2016 and by the Ministry of Education, Culture and Sport of Spain, the University of Alicante, the Generalitat Valenciana and the Spanish Government through projects TIN2015-65136-C2-2-R, TIN2015-65100-R, PROMETEOII/2014/001 and FUNDACIONBBVA2-16PREMIOI.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*. volume 10, pages 2200–2204.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18(4):467–479.

Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 1124–1128. http://www.aclweb.org/anthology/S16-1173.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear - a library for large linear classification. The Weka classifier works with version 1.33 of LIBLINEAR. http://www.csie.ntu.edu.tw/čjlin/liblinear/.

Eibe Frank, Mark A. Hall, and Ian H. Witten. 2016. The weka workbench. online appendix for "data mining: Practical machine learning tools and techniques". Technical report.

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)*. Association for Computational Linguistics, Atlanta, Georgia, USA, pages 328–332. http://www.aclweb.org/anthology/S13-2054.

Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein. 2015. Webis: An ensemble for twitter sentiment detection. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 582–589. http://www.aclweb.org/anthology/S15-2097.

Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska De Jong, and Uzay Kaymak. 2015. Exploiting emoticons in polarity classification of text. *J. Web Eng.* 14(1&2):22–40.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 168–177.

Ludmila I Kuncheva. 2004. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*. ACM, pages 24–26.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Association for Computational Linguistics, Atlanta, Georgia, USA, pages 321–327. http://www.aclweb.org/anthology/S13-2053.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 1–18. http://www.aclweb.org/anthology/S16-1001.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation*. Vancouver, Canada, SemEval '17.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 347–354.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif Mohammad. 2014. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 443–447. http://www.aclweb.org/anthology/S14-2077.