SU-FMI: System Description for SemEval-2014 Task 9 on Sentiment Analysis in Twitter

Boris Velichkov[*] **Borislav Kapukaranov**[†] **Ivan Grozev**[‡] **Jeni Karanesheva**[§] **Todor Mihaylov**[¶]
Yasen Kiprov[||] **Georgi Georgiev**[**] **Ivan Koychev**[††] **Preslav Nakov**[‡‡]

Abstract

We describe the submission of the team of the Sofia University to SemEval-2014 Task 9 on Sentiment Analysis in Twitter. We participated in *subtask B*, where the participating systems had to predict whether a Twitter message expresses positive, negative, or neutral sentiment. We trained an SVM classifier with a linear kernel using a variety of features. We used publicly available resources only, and thus our results should be easily replicable. Overall, our system is ranked 20th out of 50 submissions (by 44 teams) based on the average of the three 2014 evaluation data scores, with an F1-score of 63.62 on general tweets, 48.37 on sarcastic tweets, and 68.24 on LiveJournal messages.

1 Introduction

We describe the submission of the team of the Sofia University, Faculty of Mathematics and Informatics (SU-FMI) to SemEval-2014 Task 9 on Sentiment Analysis in Twitter (Rosenthal et al., 2014).

This SemEval challenge had two subtasks:

- *subtask A* (term-level) asks to predict the sentiment of a phrase inside a tweet;
- *subtask B* (message-level) asks to predict the overall sentiment of a tweet message.

In both subtasks, the sentiment can be positive, negative, or neutral. Here are some examples:

- *positive*: Gas by my house hit $3.39!!!! I'm going to Chapel Hill on Sat. :)
- *neutral*: New York Giants: Game-by-Game Predictions for the 2nd Half of the Season http://t.co/yK9VTjcs
- *negative*: Why the hell does Selma have school tomorrow but Parlier clovis & others don't?
- *negative (sarcastic)*: @MetroNorth wall to wall people on the platform at South Norwalk waiting for the 8:08. Thanks for the Sat. Sched. Great sense

Below we first describe our preprocessing, features and classifier in Section 2. Then, we discuss our experiments, results and analysis in Section 3. Finally, we conclude with possible directions for future work in Section 4.

2 Method

Our approach is inspired by the highest scoring team in 2013, NRC Canada (Mohammad et al., 2013). We reused many of their resources.[1]

Our system consists of two main submodules, (i) feature extraction in the framework of GATE (Cunningham et al., 2011), and (ii) machine learning using SVM with linear kernels as implemented in LIBLINEAR[2] (Fan et al., 2008).

[*] Sofia University, `bobby.velichkov@gmail.com`
[†] Sofia University, `b.kapukaranov@gmail.com`
[‡] Sofia University, `iigrozev@gmail.com`
[§] Sofia University, `j.karanesheva@gmail.com`
[¶] Sofia University, `tbmihailov@gmail.com`
[||] Sofia University, `yasen.kiprov@gmail.com`
[**] Ontotext, `g.d.georgiev@gmail.com`
[††] Sofia University, `koychev@fmi.uni-sofia.bg`
[‡‡] Qatar Computing Research Institute, `pnakov@qf.org.qa`

[1] `http://www.umiacs.umd.edu/~saif/WebPages/Abstracts/NRC-SentimentAnalysis.htm`

[2] `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 590–595,
Dublin, Ireland, August 23-24, 2014.

2.1 Preprocessing

We integrated a pipeline of various resources for tweet analysis that are already available in GATE (Bontcheva et al., 2013) such as a Twitter tokenizer, a sentence splitter, a hashtag tokenizer, a Twitter POS tagger, a morphological analyzer, and the Snowball[3] stemmer.

We further implemented in GATE some shallow text processing components in order to handle negation contexts, emoticons, elongated words, all-caps words and punctuation. We also added components to find words and phrases contained in sentiment lexicons, as well as to annotate words with word cluster IDs using the lexicon built at CMU,[4] which uses the Brown clusters (Brown et al., 1992) as implemented[5] by (Liang, 2005).

2.2 Features

2.2.1 Sentiment lexicon features

We used several preexisting lexicons, both manually designed and automatically generated:

- Minqing Hu and Bing Liu opinion lexicon (Hu and Liu, 2004): 4,783 positive and 2,006 negative terms;
- MPQA Subjectivity Cues Lexicon (Wilson et al., 2005): 8,222 terms;
- Macquarie Semantic Orientation Lexicon (MSOL) (Mohammad et al., 2009): 30,458 positive and 45,942 negative terms;
- NRC Emotion Lexicon (Mohammad et al., 2013): 14,181 terms with specified emotion.

For each lexicon, we find in the tweet the terms that are listed in it, and then we calculate the following features:

- Negative terms count;
- Positive terms count;
- Positive negated terms count;
- Positive/negative terms count ratio;
- Sentiment of the last token;
- Overall sentiment terms count.

We further used the following lexicons:

- NRC Hashtag Sentiment Lexicon: list of words and their associations with positive and negative sentiment (Mohammad et al., 2013): 54,129 unigrams, 316,531 bigrams, 480,010 pairs, and 78 high-quality positive and negative hashtag terms;
- Sentiment140 Lexicon: list of words with associations to positive and negative sentiments (Mohammad et al., 2013): 62,468 unigrams, 677,698 bigrams, 480,010 pairs;
- Stanford Sentiment Treebank: contains 239,231 evaluated words and phrases. If a word or a phrase was found in the tweet, we took the given sentiment label.

For the NRC Hashtag Sentiment Lexicon and the Sentiment140 Lexicon, we calculated the following features for unigrams, bigrams and pairs:

- Sum of positive terms' sentiment;
- Sum of negative terms' sentiment;
- Sum of the sentiment for all terms in the tweet;
- Sum of negated positive terms' sentiment;
- Negative/positive terms ratio;
- Max positive sentiment;
- Min negative sentiment;
- Max sentiment of a term.

We used different features for the two lexicon groups because their contents differ. The first four lexicons provide a discrete sentiment value for each word. In contrast, the following two lexicons offer numeric sentiment scores, which allows for different feature types such as sums and min/max scores.

Finally, we manually built a new lexicon with all emoticons we could find, where we assigned to each emoticon a positive or a negative label. We then calculated four features: number of positive and negative emoticons in the tweet, and whether the last token is a positive or a negative emoticon.

[3] http://snowball.tartarus.org/

[4] http://www.ark.cs.cmu.edu/TweetNLP/cluster_viewer.html

[5] http://github.com/percyliang/brown-cluster

2.2.2 Tweet-level features

We use the following tweet-level features:

- **All caps:** the number of words with all characters in upper case;
- **Hashtags:** the number ot hashtags in the tweet;
- **Elongated words:** the number of words with character repetitions.

2.2.3 Term-level features

We used the following term-level features:

- **Word n-grams**: presence or absence of 1-grams, 2-grams, 3-grams, 4-grams, and 5-grams. We add an `NGRAM` prefix to each n-gram. Unfortunately, the n-grams increase the feature space greatly and contribute to higher sparseness. They also slow down training dramatically. That is why our final submission only includes 1-grams.
- **Character n-grams**: presence or absence of one, two, three, four and five-character prefixes and suffixes of all words. We add a `PRE` or `SUF` prefix to each character n-gram.
- **Negations:** the number of negated contexts. We define a negated context as a segment of a tweet that starts with a negation word (e.g., *no*, *shouldnt*) from our custom gazetteer and ends with one of the punctuation marks: ,, ., :, ;, !, ?. A negated context affects the n-gram and the lexicon features: we add a `NEG` suffix to each word following the negation word, e.g., *perfect* becomes *perfect_NEG*.
- **Punctuation:** the number of contiguous sequences of exclamation marks, of question marks, of *either* exclamation or question marks, and of *both* exclamation and question marks. Also, whether the last token contains an exclamation or a question mark (excluding URLs).
- **Stemmer:** the stem of each word, excluding URLs. We add a `STEM` prefix to each stem.
- **Lemmatizer:** the lemma of each word, excluding URLs. We add a `LEMMA` prefix to each lemma. We use the built-in GATE Morphological analyser as our lemmatizer.
- **Word and word bigram clusters:** word clusters have been shown to improve the performance of supervised NLP models (Turian et al., 2010). We use the word clusters built by CMU's NLP toolkit, which were produced over a collection of 56 million English tweets (Owoputi et al., 2012) and built using the Percy Liang's HMM-based implementation[6] of Brown clustering (Liang, 2005; Brown et al., 1992), which group the words into 1,000 hierarchical clusters. We use two features based on these clusters:
  - presence/absence of a word in a word cluster;
  - presence/absence of a bigram in a bigram cluster.
- **POS tagging:** Social media are generally hard to process using standard NLP tools, which are typically developed with newswire text in mind. Such standard tools are not a good fit for Twitter messages, which are too brief, contain typos and special wordforms. Thus, we used a specialized POS tagger, TwitIE, which is available in GATE (Bontcheva et al., 2013), and which we integrated in our pipeline. It provides (i) a tokenizer specifically trained to handle smilies, user names, URLs, etc., (ii) a normalizer to correct slang and misspellings, and (iii) a POS tagger that uses the Penn Treebank tagset, but is optimized for tweets. Using the TwitIE toolkit, we performed POS tagging and we extracted all POS tag types that we can find in the tweet together with their frequencies as features.

2.3 Classifier

For classification, we used the above features and a support vector machine (SVM) classifier as implemented in LIBLINEAR. This is a very scalable implementation of SVM that does not support kernels, and is suitable for classification on large datasets with a large number of features. This is particularly useful for text classification, where the number of features is very large, which means that the data is likely to be linearly separable, and thus using kernels is not really necessary. We scaled the SVM input and we used L2-regularization during training.

[6] `https://github.com/percyliang/brown-cluster`

3 Experiments, Results, Analysis

3.1 Experimental setup

At development time, we trained on train-2013, tuned the C value of SVM on dev-2013, and evaluated on test-2013 (Nakov et al., 2013). For our submission, we trained on train-2013+dev-2013, and we evaluated on the 2014 test dataset provided by the organizers. This dataset contains two parts and a total of five datasets: (a) progress test (the Twitter and SMS test datasets for 2013), and (b) new test datasets (from Twitter, from Twitter with sarcasm, and from LiveJournal). We used C=0.012, which was best on development.

3.2 Official results

Due to our very late entering in the competition, we have only managed to perform a small number of experiments, and we only participated in subtask B. We were ranked 20th out of 50 submissions; our official results are shown in Table 1. The numbers after our score are the delta to the best solution. We have also included a ranking among 2014 participant systems on the 2013 data sets, released by the organizers.

Data Category	F1-score (best)	Ranking
tweets2014	63.62 (6.23)	23
sarcasm2014	48.34 (9.82)	19
LiveJournal2014	68.23 (6.60)	21
tweets2013	60.96 (9.79)	29
SMS2013	61.67 (8.61)	16
2014 mean	60.07 (7.55)	20

Table 1: Our submitted system for subtask B.

3.3 Analysis

Tables 2 and 3 analyze the impact of the individual features. They show the F1-scores and the loss when a feature or a group of features is removed; we show the impact on all test datasets, both from 2013 and from 2014. The exception here is the *all + ngrams* row, which contains our scores if we had used the n-grams feature group.

The features are sorted by their impact on the Twitter2014 test set. We can see that the three most important feature groups are POS tags, word/bigram clusters, and lexicons.

We can further see that although the overall lexicon feature group is beneficial, some of the lexicons actually hurt the 2014 score and we would have been better off without them. These are the Sentiment140 lexicon, the Stanford Sentiment Treebank and the NRC Emotion lexicon. The highest gain we get is from the lexicons of Minqing Hu and Bing Liu. It must be noted that using lexicons with good results apparently depends on the context, e.g., the Sentiment140 lexicon seems to be helping a lot with the LiveJournal test dataset, but it hurts the Sarcasm score by a sizeable margin.

Another interesting observation is that even though including the n-gram feature group is performing notably better on the Twitter2013 test dataset, it actually worsens performance on all 2014 test sets. Had we included it in our results, we would have scored lower.

The negation context feature brings little in regards to regular tweets or LiveJournal text, but it heavily improves our score on the Sarcasm tweets.

It is unclear why our results differ so much from those of the NRC-Canada team in 2013 since our features are quite similar. We attribute the difference to the fact that some of the lexicons we use actually hurt our score as we mentioned above. Another difference could be that last year's NRC system uses n-grams, which we have disabled as they lowered our scores. Last but not least, there could be bugs lurking in our feature representation that additionally lower our results.

3.4 Post-submission improvements

First, we did more extensive experiments to validate our classifier's C value. We found that the best value for C is actually 0.08 instead of our original proposal 0.012.

Then, we experimented further with our lexicon features and we removed the following ones, which resulted in significant improvement over our submitted version:

- Sentiment of the last token for NRC Emotion, MSOL, MPQA, and Bing Liu lexicons;
- Max term positive, negative and sentiment scores for unigrams of Sentiment140 and NRC Sentiment lexicons;
- Max term positive, negative and sentiment scores for bigrams of Sentiment140 and NRC Sentiment lexicons;
- Max term positive, negative and sentiment scores for hashtags of Sentiment140 and NRC Sentiment lexicons.

Feature Diff	SMS2013	SMS2013 delta	Twitter2013	Twitter2013 delta
submitted features	**61.67**		**60.96**	
no POS tags	54.73	-6.94	52.32	-8.64
no word clusters	58.06	-3.61	55.44	-5.52
all lex removed	59.94	-1.73	58.35	-2.61
no Hu-Liu lex	60.56	-1.11	60.10	-0.86
all + ngrams	61.37	-0.30	62.22	1.26
no NRC #lex	61.35	-0.32	60.66	-0.30
no MSOL lex	61.88	0.21	61.35	0.39
no Stanford lex	61.84	0.17	61.02	0.06
no negation cntx	61.94	0.27	60.88	-0.08
no encodings	61.74	0.07	60.92	-0.04
no NRC emo lex	61.67	0.00	60.96	0.00
no Sent140 lex	61.61	-0.06	60.32	-0.64

Table 2: Ablation experiments on the 2013 test sets.

Feature Diff	LiveJournal	LJ delta	Twitter	Twitter delta	Sarcasm	Sarcasm delta
submitted features	**68.23**		**63.62**		**48.34**	
no POS tags	62.28	-5.95	59.00	-4.62	43.70	-4.64
no word clusters	65.08	-3.15	59.82	-3.80	43.96	-4.38
all lex removed	66.16	-2.07	60.73	-2.89	49.59	1.25
no Hu-Liu lex	66.44	-1.79	62.15	-1.47	46.72	-1.62
all + ngrams	67.79	-0.44	62.96	-0.66	47.82	-0.52
no NRC #lex	66.81	-1.42	63.25	-0.37	47.54	-0.80
no MSOL lex	68.50	0.27	63.54	-0.08	48.34	0.00
no Stanford lex	67.86	-0.37	63.70	0.08	48.34	0.00
no negation cntx	68.09	-0.14	63.62	0.00	46.37	-1.97
no encodings	68.23	0.00	63.64	0.02	47.54	-0.80
no NRC emo lex	68.24	0.01	63.62	0.00	48.34	0.00
no Sent140 lex	67.32	-0.91	63.94	0.32	49.47	1.13

Table 3: Ablation experiments on the 2014 test sets.

The improved scores are shown in Table 4, with the submitted and the best system results.

Test Set	New F1	Old F1	Best
tweets2014	66.23	63.62	69.85
sarcasm2014	50.00	48.34	58.16
LiveJournal2014	69.41	68.24	74.84
tweets2013	63.08	60.96	70.75
SMS2013	62.28	61.67	70.28
2014 mean	62.20	60.07	67.62

Table 4: Our post-submission results.

4 Conclusion and Future Work

We have described the system built by the team of SU-FMI for SemEval-2014 task 9. Due to our late entering in the competition, we were only ranked 20th out of 50 submissions (from 44 teams).

We have made some interesting observations about the impact of the different features. Among the best-performing feature groups were POS-tag counts, word cluster presence and bigrams, the Hu-Liu lexicon and the NRC Hashtag Sentiment lexicon. These had the most sustainable performance over the 2013 and the 2014 test datasets. Others we did not use, seemingly more context dependent, seem to have been more suited for the 2013 test sets like the n-grams feature group.

Even though we made some improvements after submitting our initial version, we feel there is more to gain and optimize. There seem to be several low-hanging fruits based on our experiments data, which could add few points to our F1-scores.

Going forward, our goal is to extend our experiments with more feature sub- and super-sets and to turn our classifier into a state-of-the-art performer.

Acknowledgments

This work is partially supported by the FP7-ICT Strategic Targeted Research Project PHEME (No. 611233), and by the European Social Fund through the Human Resource Development Operational Programme under contract BG051PO001-3.3.06-0052 (2012/2014).

References

Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark Greenwood, Diana Maynard, and Niraj Aswani. 2013. TwitIE: An open-source information extraction pipeline for microblog text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '13, pages 83–90, Hissar, Bulgaria.

Peter Brown, Peter deSouza, Robert Mercer, Vincent Della Pietra, and Jenifer Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

Hamish Cunningham, Diana Maynard, and Kalina Bontcheva. 2011. *Text Processing with GATE*. Gateway Press CA.

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.

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*, KDD '04, pages 168–177, New York, NY, USA.

Percy Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Saif Mohammad, Cody Dunne, and Bonnie Dorr. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Volume 2*, EMNLP '09, pages 599–608, Singapore.

Saif 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 '13, Atlanta, Georgia, USA.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval '13, pages 312–320, Atlanta, Georgia, USA.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, and Nathan Schneider. 2012. Part-of-speech tagging for Twitter: Word clusters and other advances. Technical Report CMU-ML-12-107, Carnegie Mellon University.

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, Dublin, Ireland.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Uppsala, Sweden.

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*, HLT '05, pages 347–354, Vancouver, British Columbia, Canada.