RTRGO: Enhancing the GU-MLT-LT System for Sentiment Analysis of Short Messages

Tobias Günther
Retresco GmbH
retresco.de
email@tobias.io

Jean Vancoppenolle
ferret go GmbH
ferret-go.com
jean.vcop@gmail.com

Richard Johansson
University of Gothenburg
www.svenska.gu.se
richard.johansson@gu.se

Abstract

This paper describes the enhancements made to our GU-MLT-LT system (Günther and Furrer, 2013) for the SemEval-2014 re-run of the SemEval-2013 shared task on sentiment analysis in Twitter. The changes include the usage of a Twitter-specific tokenizer, additional features and sentiment lexica, feature weighting and random subspace learning. The improvements result in an increase of 4.18 F-measure points on this year's Twitter test set, ranking 3rd.

1 Introduction

Automatic analysis of sentiment expressed in text is an active research area in natural language processing with obvious commercial interest. In the simplest formulation of the problem, sentiment analysis is framed as a categorization problem over documents, where the set of categories is typically a set of polarity values, such as positive, neutral, and negative. Many approaches to document-level sentiment classification have been proposed. For an overview see e.g. Liu (2012).

Text in social media and in particular microblog messages are a challenging text genre for sentiment classification, as they introduce additional problems such as short text length, spelling variation, special tokens, topic variation, language style and multilingual content. Following Pang et al. (2002), most sentiment analysis systems have been based on standard text categorization techniques, e.g. training a classifier using some sort of bag-of-words feature representation. This is also true for sentiment analysis of microblogs. Among the first to work specifically with Twitter[1] data were Go et al. (2009), who use emoticons as labels for the messages. Similarly, Davidov et al. (2010), Pak and Paroubek (2010), and Kouloumpis et al. (2011) use this method of distant supervision to overcome the data acquisition barrier. Barbosa and Feng (2010) make use of three different sentiment detection websites to label messages and use mostly non-lexical features to improve the robustness of their classifier. Bermingham and Smeaton (2010) investigate the impact of the shortness of Tweets on sentiment analysis and Speriosu et al. (2011) propagate information from seed labels along a linked structure that includes Twitter's follower graph. There has also been work on lexicon-based approaches to sentiment analysis of microblogs, such as O'Connor et al. (2010), Thelwall et al. (2010) and Zhang et al. (2011). For a detailed discussion see Günther (2013).

In 2013, the International Workshop on Semantic Evaluation (SemEval) organized a shared task on sentiment analysis in Twitter (Nakov et al., 2013) to enable a better comparison of different approaches for sentiment analysis of microblogs. The shared task consisted of two subtasks: one on recognizing contextual polarity of a given subjective expression (Task A), and one on document-level sentiment classification (Task B). For both tasks, the training sets consisted of manually labeled Twitter messages, while the test sets consisted of a Twitter part and an SMS part in order to test domain sensitivity. Among the best performing systems were Mohammad et al. (2013), Günther and Furrer (2013) and Becker et al. (2013), who all train linear models on a variety of task-specific features. In this year the corpus resources were used for a re-run of the shared task (Rosenthal et al., 2014), introducing two new Twitter test sets, as well as LiveJournal data.

[1] A popular microblogging service on the internet, its messages are commonly referred to as "Tweets."

2 System Desciption

This section describes the details of our sentiment analysis system, focusing on the differences to our last year's implementation. This year we only participated in the subtask on whole message polarity classification (Subtask B).

2.1 Preprocessing

For **tokenization** of the messages we use the tokenizer of Owoputi et al. (2013)'s Twitter NLP Tools[2], which include a tokenizer and part-of-speech tagger optimized for the usage with Tweets. The tokenizer contains a regular expression grammar for recognizing emoticons, which is an especially valuable property in the context of sentiment analysis due to the high emotional expressiveness of emoticons.

It is well known that the way word tokens are represented may have a significant impact on the performance of a lexical classifier. This is particularly true in natural language processing of social media, where we run into the problem of spelling variation causing extreme lexical sparsity. To deal with this issue we **normalize** the tokens with the following technique: First, all tokens are converted to lowercase and the hashtag sign (#) is removed if present. If the token is not present in an English word list or any of the used sentiment lexica (see below), we remove all directly repeated letters after the first repetition (e.g. greeeeaaat → greeaat). If the resulting token is still not present in any of the lexical resources, we allow no direct repetition of letters at all. While this might lead to lexical collisions in some cases (e.g. goooodd → goodd → god), it is an easy and efficient way to remove some lexical sparsity. While generating all possible combinations of deletions and checking the resulting tokens against a lexical resource is another option, a correct disambiguation of the intended word would require a method making use of context knowledge (e.g. goooodd → good, vs. goooodd → god).

2.2 Features

We use the following set of features as input to our supervised classifier:

- The normalized tokens as **unigrams** and **bigrams**, where stopword and punctuation tokens are excluded from bigrams
- The **word stems** of the normalized tokens, reducing inflected forms of a word to a common form. The stems were computed using the Porter stemmer algorithm (Porter, 1980)
- The IDs of the token's **word clusters**. The clusters were generated by performing Brown clustering (Brown et al., 1992) on 56,345,753 Tweets by Owoputi et al. (2013) and are available online.[2]
- The presence of a hashtag or URL in the message (one feature each)
- The presence of a question mark token in the message
- We use the opinion lexicon by Bing Liu (Hu and Liu, 2004), the MPQA subjectivity lexicon (Wiebe et al., 2005) and the Twitrratr wordlist, which all provide a list of positive and negative words, to compute a prior polarity of the message. For each of the three **sentiment lexica** two features capture whether the majority of the tokens in the message were in the positive or negative sentiment list. The same is done for hashtags using the NRC hashtag sentiment lexicon (Mohammad et al., 2013).
- We apply special handling to features in a **negation** context. A token is considered as negated if it occurs after a negation word (up to the next punctuation). All token, stem and word cluster features are marked with a negation prefix. Additionally, the polarity for token in a negation context is inverted when computing the prior lexicon polarity.
- We use the **part-of-speech tags** computed by the part-of-speech tagger of the Twitter NLP tools by Owoputi et al. (2013) to **exclude** certain tokens. Assuming they do not carry any helpful sentiment information, no features are computed for token recognized as name (tag ^) or user mention (tag @).
- We also employ **feature weighting** to give more importance to certain features and indication of **emphasis** by the author. Normally, all features described above receive weight 1 if they are present and weight 0 if they are absent. For each of the following cases we add +1 to the weight of a token's unigram, stem and word cluster features:

[2] `http://www.ark.cs.cmu.edu/TweetNLP`

  - The original (not normalized) token is all uppercase
  - The original token has more than three adjacent repetitions of one letter
  - The token is an adjective or emoticon (according to its part-of-speech tag)

  Furthermore, the score of each token is divided in half, if the token occurs in a **question context**. A token is considered to be in a question context, if it occurs before a question mark (up to the next punctuation).

2.3 Machine Learning Methods

All training was done using the open-source machine learning toolkit *scikit-learn*[3] (Pedregosa et al., 2011). Just as in our last year's system we trained linear one-versus-all classifiers using stochastic gradient descent optimization with hinge loss and elastic net regularization.[4] For further details see Günther and Furrer (2013). The number of iterations was set to 1000 for the final model and 100 for the experiments.

It is widely observed that training on a lot of lexical features can lead to brittle NLP systems, that are easily overfit to particular domains. In social media messages the brittleness is particularly acute due to the wide variation in vocabulary and style. While this problem can be eased by using corpus-induced word representations such as the previously introduced word cluster features, it can also be addressed from a learning point of view. Brittleness can be caused by the problem that very strong features (e.g. emoticons) drown out the effect of other useful features.

The method of **random subspace learning** (Søgaard and Johannsen, 2012) seeks to handle this problem by forcing learning algorithms to produce models with more redundancy. It does this by randomly corrupting training instances during learning, so if some useful feature is correlated with a strong feature, the learning algorithm has a better chance to assign it a nonzero weight. We implemented random subspace learning by training the classifier on a concatenation of 25 corrupted copies of the training set. In a corrupted copy, each feature was randomly disabled with a probability of 0.2. Just as for the classifier, the hyperparameters were optimized empirically.

3 Experiments

For the experiments and the training of the final model we used the joined training and development sets of subtask B. We were able to retrieve 10368 Tweets, of which we merged all samples labeled as objective into the neutral class. This resulted in a training set of 3855 positive, 4889 neutral and 1624 negative tweets. The results of the experiments were obtained by performing 10-fold cross-validation, predicting positive, negative and neutral class. Just as in the evaluation of the shared task the results are reported as average F-measure (F_1) between positive and negative class.

To be able to evaluate the contribution of the different features groups to the final model we perform an ablation study. By disabling one feature group at the time one can easily compare the performance of the model without a certain feature to the model using the complete feature set. In Table 1 we present the results for the feature groups bigrams (2gr), stems (stem), word clusters (wc), sentiment lexica (lex), negation (neg), excluding names and user mentions (excl), feature weighting (wei) and random subspace learning (rssl).

	Negative		**Positive**		**Avg.**
	Prec	**Rec**	**Prec**	**Rec**	**F_1**
ALL	54.80	71.67	76.70	75.41	69.08
-2gr	-0.55	-0.49	-0.35	+0.20	-0.31
-stem	-1.47	-1.72	-0.49	-0.03	-0.92
-wc	-1.45	-1.60	-0.40	-1.66	-1.29
-lex	-1.73	**-5.11**	+1.06	-2.75	**-1.99**
-neg	**-1.90**	-3.14	**-1.30**	+0.36	-1.43
-excl	+0.31	-0.99	+0.59	+0.08	+0.08
-wei	-1.57	+0.43	-0.84	-0.34	-0.73
-rssl	+2.04	-4.37	+1.38	**-2.88**	-0.67

Table 1: Feature ablation study

Looking at Table 1, we can see that removing the sentiment lexica features causes the biggest drop in performance. This is especially true for the recall of the negative class, which is underrepresented in the training data and can thus profit the most from prior domain knowledge. When comparing to the features of our last year's system, it becomes clear that the used sentiment lexica can provide a much bigger gain in performance than the previously used SentiWordNet. Even though they are outperformed by the sentiment lexica, the word cluster features still provide an additional in-

[3] Version 0.13.1, `http://scikit-learn.org`.

[4] `SGDClassifier(penalty='elasticnet', alpha=0.001, l1_ratio=0.85, n_iter=1000, class_weight='auto')`

	GU-MLT-LT (2013)			RTRGO (2014)		
	F_1 pos/neg	F_1 3-class	Accuracy	F_1 pos/neg	F_1 3-class	Accuracy
Twitter2013	65.42	68.13	70.42	**69.10**	70.92	72.54
Twitter2014	65.77	66.59	69.40	**69.95**	69.99	72.53
SMS2013	62.65	66.93	69.09	**67.51**	72.15	75.54
LiveJournal2014	68.97	68.42	68.39	**72.20**	72.29	72.33
Twitter2014Sarcasm	54.11	56.91	58.14	**47.09**	49.34	51.16

Table 2: Final results of our submissions on the different test sets (Subtask B)

crease in performance and can, in contrast to sentiment lexica, be learned in a completely unsupervised manner. Negation handling is an important feature to boost the precision of the classifier, while using random subspace learning increases the recall of the classes, which indicates that the technique indeed leads to more redundant models.

Another interesting question in sentiment analysis is, how machine learning methods compare to simple methods only relying on sentiment wordlists and how much training data is needed to outperform them. Figure 1 shows the results of a training size experiment, in which we tested classifiers, trained on different portions of a training set, on the same test set (10-fold cross validated). The two horizontal lines indicate the performance of two simple classifiers, using the Twitrratr wordlist (359 entries, labeled TRR) or Bing Liu opinion lexicon (6789 entries, labeled LIU) with a simple majority-vote strategy (choosing the neutral class in case of no hits or no majority and including a polarity switch for token in a negation context). The baseline of the machine learning classifiers is a logistic regression classifier using only uni- and bigram features and negation handling (labeled BOW). To this baseline we add either the lexicon features for the Bing Liu opinion lexicon and the Twitrratr wordlist (labeled +LEX) or all other features described in section 2.2 excluding lexicon features (labeled +REST). Looking at the results, we can see that a simple bag of words classifier needs about 250 samples of each class to outperform the TRR list and about 700 samples of each class to outperform the LIU lexicon on the common test set. Adding the features that can be obtained without having sentiment lexica available (+REST) reduces the needed training samples about half. It is worth noting that from a training set size of 1250 samples per class the +REST-classifier is able to match the results of the classifier combining bag of words and lexicon features (+LEX).

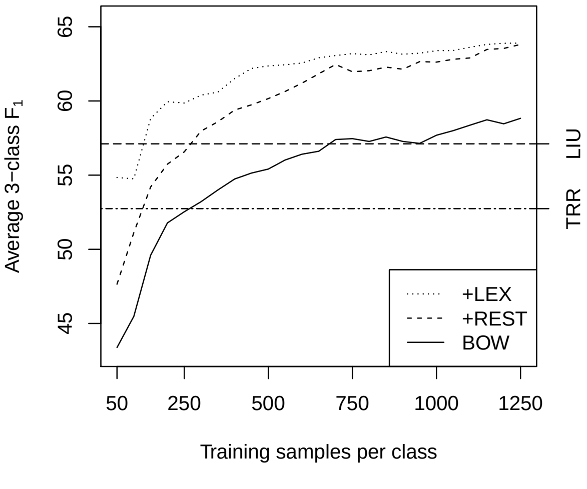

Figure 1: Training size experiment

4 Results and Conclusion

The results of our system are presented in Table 2, where the bold column marks the results relevant to our submission to this year's shared task. We also give results for our last year's system. Beside the average F-measure between positive and negative class, on which the shared task is evaluated, we also provide the results of both systems as average F-measure over all three classes and accuracy to create possibilities for better comparison to other research. In this paper we showed several ways to improve a machine learning classifier for the use of sentiment analysis in Twitter. Compared to our last year's system we were able to increase the performance about several F-measure points on all non-sarcastic datasets.

Acknowledgements

We would like to thank the organizers of the shared task for their effort, as well as the anonymous reviewers for their helpful comments on the paper.

References

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44. Association for Computational Linguistics.

Lee Becker, George Erhart, David Skiba, and Valentine Matula. 2013. Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 333–340, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Adam Bermingham and Alan F Smeaton. 2010. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1833–1836. ACM.

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.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 241–249. Association for Computational Linguistics.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford.*

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)*, pages 328–332, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Tobias Günther. 2013. Sentiment analysis of microblogs. Master's thesis, University of Gothenburg, June.

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*, pages 168–177. ACM.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 538–541.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

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-2013)*, Atlanta, Georgia, USA, June.

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 2013)*, pages 312–320, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R Routledge, and Noah A Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 122–129.

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. In *Proceedings of NAACL 2013*.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC*, volume 2010.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanova. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, August.

Anders Søgaard and Anders Johannsen. 2012. Robust learning in random subspaces: Equipping NLP for OOV effects. In *COLING (Posters)*, pages 1171–1180.

Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Ley Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. 2011. Combining lexiconbased and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011-89*.