ASVUniOfLeipzig: Sentiment Analysis in Twitter using Data-driven Machine Learning Techniques

Robert Remus
Natural Language Processing Group,
Department of Computer Science,
University of Leipzig, Germany
rremus@informatik.uni-leipzig.de

Abstract

This paper describes University of Leipzig's approach to SemEval-2013 task 2B on Sentiment Analysis in Twitter: message polarity classification. Our system is designed to function as a baseline, to see what we can accomplish with well-understood and purely data-driven lexical features, simple generalizations as well as standard machine learning techniques: We use one-against-one Support Vector Machines with asymmetric cost factors and linear "kernels" as classifiers, word uni- and bigrams as features and additionally model negation of word uni- and bigrams in word n-gram feature space. We consider generalizations of URLs, user names, hash tags, repeated characters and expressions of laughter. Our method ranks 23 out of all 48 participating systems, achieving an averaged (positive, negative) F-Score of 0.5456 and an averaged (positive, negative, neutral) F-Score of 0.595, which is above median and average.

1 Introduction

In SemEval-2013's task 2B on *Sentiment Analysis in Twitter*, given a Twitter message, i.e. a tweet, the goal is to classify whether this tweet is of positive, negative, or neutral polarity (Wilson et al., 2013), i.e. the task is a ternary polarity classification.

Due to Twitter's growing popularity, the availability of large amounts of data that go along with that and the fact, that many people freely express their opinion on virtually everything using Twitter, research on sentiment analysis in Twitter has received a lot of attention lately (Go et al., 2009; Pak and Paroubek, 2010). Language is usually used casually in Twitter and exhibits interesting properties. Therefore, some studies specifically address certain issues, e.g. a tweet's length limitation of 140 characters, some studies leverage certain language characteristics, e.g. the presence of emoticons etc.

Davidov et al. (2010) identify various "sentiment types" defined by Twitter hash tags (e.g. #bored) and smileys (e.g. :S) using words, word n-grams, punctuation marks and patterns as features. Barbosa and Feng (2010) map words to more general representations, i.e. part of speech (POS) tags and the words' prior subjectivity and polarity. Additionally, they count the number of re-tweets, hash tags, replies, links etc. They then combine the outputs of 3 online sources of labeled but noisy and biased Twitter data into a more robust classification model. Saif et al. (2012) also address data sparsity via word clustering methods, i.e. semantic smoothing and sentiment-topics extraction. Agarwal et al. (2011) contrast a word unigram model, a tree kernel model and a model of various features, e.g. POS tag counts, summed up prior polarity scores, presence or absence of capitalized text, all applied to binary and ternary polarity classification. Kouloumpis et al. (2011) show that Twitter-specific feature engineering, e.g. representing the presence or absence of abbreviations and character repetitions improves model quality. Jiang et al. (2011) focus on target-dependent polarity classification regarding a given user query.

While various models and features have been proposed, word n-gram models proved to be competitive in many studies (Barbosa and Feng, 2010; Agar-

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 450–454, Atlanta, Georgia, June 14-15, 2013. ©2013 Association for Computational Linguistics

wal et al., 2011; Saif et al., 2012) yet are straightforward to implement. Moreover, word n-gram models do not rely on hand-crafted and generally $\{$genre, domain$\}$-non-specific resources, e.g. prior polarity dictionaries like *SentiWordNet* (Esuli and Sebastiani, 2006) or *Subjectivity Lexicon* (Wiebe et al., 2005). In contrast, purely data-driven word n-gram models are *domain-specific* per se: they "let the data speak for themselves". Therefore we believe that carefully designing such a baseline using well-understood and purely data-driven lexical features, simple generalizations as well as standard machine learning techniques is a worthwhile endeavor.

In the next Section we describe our system. In Section 3 we discuss its results in SemEval-2013 task 2B and finally conclude in Section 4.

2 System Description

We approach the ternary polarity classification via one-against-one (Hsu and Lin, 2002) Support Vector Machines (SVMs) (Vapnik, 1995; Cortes and Vapnik, 1995) using a linear "kernel" as implemented by *LibSVM*[1]. To deal with the imbalanced class distribution of positive $(+)$, negative $(-)$ and neutral-or-objective (0) instances, we use asymmetric cost factors C_+, C_-, C_0 that allow for penalizing false positives and false negatives differently inside the one-against-one SVMs. While the majority class' C_0 is set to 1.0, the minority classes' $C_{\{+,-\}}$s are set as shown in (1)

$$C_{\{+,-\}} = \frac{\text{\#(0-class instances)}}{\text{\#(\{+, −\}-class instances)}} \qquad (1)$$

similar to Morik et al. (1999)'s suggestion.

2.1 Data

To develop our system, we use all training data available to us for training and all development data available to us for testing, after removing 75 duplicates from the training data and 2 duplicates from the development data. Please note that 936 tweets of the originally provided training data and 3 tweets of the originally provided development data were not available at our download time[2]. Table 1 summarizes the used data's class distribution after duplicate removal.

Data	$+$	$-$	0	Σ
Training	3,263	1,278	4,132	8,673
Development	384	197	472	1,053
Σ	3,647	1,475	4,604	9,726

Table 1: Class distribution of positive $(+)$, negative $(-)$ and neutral-or-objective (0) instances in training and development data after duplicate removal.

For sentence segmentation and tokenization of the data we use *OpenNLP*[3]. An example tweet of the provided training data is shown in (1):

(1) #nacamam @naca you have to try Skywalk Deli on the 2nd floor of the Comerica building on Monroe! #bestlunche http://instagr.am/p/Rfv-RfTI-3/.

2.2 Model Selection

To select an appropriate model, we experiment with different feature sets (cf. Section 2.2.1) and different combinations of generalizations (cf. Section 2.2.2).

2.2.1 Features

We consider the following feature sets:

a. word unigrams

b. word unigrams plus negation modeling for word unigrams

c. word uni- and bigrams

d. word uni- and bigrams plus negation modeling for word unigrams

e. word uni- and bigrams plus negation modeling for word uni- and bigrams

Word uni- and bigrams are induced data-driven, i.e. directly extracted from the textual data. We perform no feature selection; neither stop words nor punctuation marks are removed. We simply encode the presence or absence of word n-grams.

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[2] Training data was downloaded on February 21, 2013, 9:18 a.m. and development data was downloaded on February 28, 2013, 10:41 a.m. using the original download script.

[3] `http://opennlp.apache.org`

Whether a word uni- or bigram is negated, i.e. appears inside of a negation scope (Wiegand et al., 2010), is detected by *LingScope*[4] (Agarwal and Yu, 2010), a state-of-the-art negation scope detection based on Conditional Random Fields (Lafferty et al., 2001). We model the negation of word n-grams in an augmented word n-gram feature space as detailedly described in Remus (2013): In this feature space, each word n-gram is either represented as present ($[1, 0]$), absent ($[0, 0]$), present inside a negation scope ($[0, 1]$) and present both inside and outside a negation scope ($[1, 1]$).

We trained a model for each feature set and chose the one that yields the highest accuracy: word uni- and bigrams plus negation modeling for word uni- and bigrams.

2.2.2 Generalizations

To account for Twitter's typical language characteristics, we consider all possible combinations of generalizations of the following character sequences, inspired by (Montejo-Ráez et al., 2012):

a. User names, that mark so-called mentions in a Tweet, expressed by `@username`.

b. Hash tags, that mark keywords or topics in a Tweet, expressed by `#keyword`.

c. URLs, that mark links to other web pages.

d. Twitpic URLs, that mark links to pictures hosted by `twitpic.com`.

e. Repeated Characters, e.g. `woooow`. We collapse characters re-occuring more than twice, e.g. `woooow` is replaced by `woow`.

f. Expressions of laughter, e.g. `hahaha`. We generalize derivatives of the "base forms" `haha`, `hehe`, `hihi` and `huhu`. A derivative must contain the base form and may additionally contain arbitrary uppercased and lowercased letters at its beginning and its end. We collapse these derivatives. E.g., `hahahah` and `HAHAhaha` and `hahaaa` are all replaced by their base form `haha`, `eheheh` and `heheHE` are all replaced by `hehe` etc.

User names, hash tags, URLs and Twitpic URLs are generalized by either simply removing them (mode I) or by replacing them with a single unique token (mode II), i.e. by forming an equivalence class. Repeated characters and expressions of laughter are generalized by collapsing them as described above.

There are $1 + \sum_{k=1}^{6} \binom{6}{k} = 64$ possible combinations of generalizations including no generalization at all. We trained a word uni- and bigram plus negation modeling for word uni- and bigrams model (cf. Section 2.2.1) for each combination and both mode I and mode II and chose the one that yields the highest accuracy: Generalization of URLs (mode I), repeated characters and expressions of laughter.

Although it may appear counterintuitive not to generalize hash tags and user names, the training data contains several re-occuring hash tags, that actually convey sentiment, e.g. `#love`, `#cantwait`, `#excited`. Similarly, the training data contains several re-occuring mentions of "celebrities", that may hint at sentiment which is usually associated with them, e.g. `@justinbieber` or `@MittRomney`.

3 Results & Discussion

To train our final system, we use all available training and development data (cf. Table 1). The SVM's "base" cost factor C is optimized via 10-fold cross validation, where in each fold 9/10th of the available data are used for training, the remaining 1/10th is used for testing. C values are chosen from $\{2 \cdot 10^{-3}, 2 \cdot 10^{-2}, 2 \cdot 10^{-1}, 2 \cdot 10^{0}, 2 \cdot 10^{1}, 2 \cdot 10^{2}, 2 \cdot 10^{3}\}$. Internally, the asymmetric cost factors C_+, C_-, C_0 (cf. Section 2) are then set to $C_{\{+,-,0\}} := C \cdot C_{\{+,-,0\}}$.

The final system is then applied to both Twitter and SMS test data (cf. Table 2). Please note that we only participate in the *constrained* setting of SemEval-2013 task 2B (Wilson et al., 2013) as we did not use any additional training data.

Test Data	$+$	$-$	0	Σ
Twitter	1,572	601	1,640	3,813
SMS	492	394	1,208	2,094

Table 2: Class distribution of positive ($+$), negative ($-$) and neutral-or-objective (0) instances in Twitter and SMS testing data.

[4] `http://sourceforge.net/projects/lingscope/`

Detailed evaluation results on Twitter test data are shown in Table 3, results on SMS test data are shown in Table 4. The ranks we achieved in the constrained only-ranking and the full constrained and unconstrained-ranking are shown in Table 5.

Class	P	R	F
$+$	0.7307	0.5833	0.6487
$-$	0.5795	0.3577	0.4424
0	0.6072	0.8098	0.6940
$+,-$	0.6551	0.4705	*0.5456*
$+,-,0$	0.6391	0.5836	0.5950

Table 3: Precision P, Recall R and F-Score F of University of Leipzig's approach to SemEval-2013 task 2B on Twitter test data distinguished by classes ($+$, $-$, 0) and averages of $+,-$ and $+,-,0$.

Class	P	R	F
$+$	0.5161	0.5854	0.5486
$-$	0.5174	0.3020	0.3814
0	0.7289	0.7881	0.7574
$+,-$	0.5168	0.4437	*0.4650*
$+,-,0$	0.5875	0.5585	0.5625

Table 4: Precision P, Recall R and F-Score F of University of Leipzig's approach to SemEval-2013 task 2B on SMS test data distinguished by classes ($+$, $-$, 0) and averages of $+,-$ and $+,-,0$.

Test data	Constr.	Un/constr.
Twitter	18 of 35	23 of 48
SMS	20 of 28	31 of 42

Table 5: Ranks of University of Leipzig's approach to SemEval-2013 task 2B on Twitter and SMS test data in the constrained only (Constr.) and the constrained and unconstrained setting (Un/constr.).

On Twitter test data our system achieved an averaged $(+,-)$ F-Score of 0.5456, which is above the average (0.5382) and above the median (0.5444). Our system ranks 23 out of 48 participating systems in the full constrained and unconstrained-ranking. Averaging over over $+,-,0$ it yields an F-Score of 0.595.

On SMS test data our system performs quite poorly compared to other participating systems as (i) we did not adapt our model to the SMS data at all, e.g. we did not consider more appropriate or other generalizations, and (ii) its class distribution is quite different from our training data (cf. Table 1 vs. 2). Our system achieved an averaged $(+,-)$ F-Score of 0.465, which is below the average (0.5008) and below the median (0.5060). Our system ranks 31 out of 42 participating systems in the full constrained and unconstrained-ranking. Averaging over over $+,-,0$ it yields an F-Score of 0.5625.

4 Conclusion

We described University of Leipzig's contribution to SemEval-2013 task 2B on Sentiment Analysis in Twitter. We approached the message polarity classification via well-understood and purely data-driven lexical features, negation modeling, simple generalizations as well as standard machine learning techniques. Despite being designed as a baseline, our system ranks midfield on both Twitter and SMS test data.

As even the state-of-the-art system achieves $(+,-)$ averaged F-Scores of 0.6902 and 0.6846 on Twitter and SMS test data, respectively, polarity classification of tweets and short messages still proves to be a difficult task that is far from being solved. Future enhancements of our system include the use of more data-driven features, e.g. features that model the distribution of abbreviations, punctuation marks or capitalized text as well as fine-tuning our generalization mechanism, e.g. by (i) generalizing only low-frequency hash tags and usernames, but not generalizing high-frequency ones, (ii) generalizing acronyms that express laughter, such as `lol` ("laughing out loud") or `rofl` ("rolling on the floor laughing").

References

S. Agarwal and H. Yu. 2010. Biomedical negation scope detection with conditional random fields. *Journal of the American Medical Informatics Association*, 17(6):696–701.

A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media (LSM*, pages 30–38.

L. Barbosa and J. Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *Proceed-*

ings of the 23rd International Conference on Computational Linguistics (COLING), pages 36–44.

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

D. Davidov, O. Tsur, and A. Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 241–249.

A. Esuli and F. Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 417–422.

A. Go, R. Bhayani, and L. Huang. 2009. Twitter sentiment classification using distant supervision. CS224N project report, Stanford University.

C. Hsu and C. Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 151–160.

E. Kouloumpis, T. Wilson, and J. Moore. 2011. Twitter sentiment analysis: The good the bad and the OMG. In *Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM)*, pages 538–541.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 282–289.

A. Montejo-Ráez, E. Martınez-Cámara, M.T. Martın-Valdivia, and L.A. Urena-López. 2012. Random walk weighting over sentiwordnet for sentiment polarity detection on twitter. In *Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 3–10.

K. Morik, P. Brockhausen, and T. Joachims. 1999. Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, pages 268–277.

A. Pak and P. Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*.

R. Remus. 2013. Negation modeling in machine learning-based sentiment analysis. In *forthcoming*.

H. Saif, Y. He, and H. Alani. 2012. Alleviating data sparsity for twitter sentiment analysis. In *Proceedings of the 2nd Workshop on Making Sense of Microposts (#MSM)*.

V. Vapnik. 1995. *The Nature of Statistical Learning*. Springer New York, NY.

J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 1(2):165–210.

M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo. 2010. A survey on the role of negation in sentiment analysis. In *Proceedings of the 2010 Workshop on Negation and Speculation in Natural Language Processing (NeSp-NLP)*, pages 60–68.

T. Wilson, Z. Kozareva, P. Nakov, A. Ritter, S. Rosenthal, and V. Stoyanov. 2013. SemEval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*.