UKPDIPF: A Lexical Semantic Approach to Sentiment Polarity Prediction in Twitter Data

Lucie Flekova[†‡], Oliver Ferschke[†‡] and Iryna Gurevych[†‡]
[†] Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Computer Science Department, Technische Universität Darmstadt
[‡] Ubiquitous Knowledge Processing Lab (UKP-DIPF)
German Institute for Educational Research
http://www.ukp.tu-darmstadt.de

Abstract

We present a sentiment classification system that participated in the SemEval 2014 shared task on sentiment analysis in Twitter. Our system expands tokens in a tweet with semantically similar expressions using a large novel distributional thesaurus and calculates the semantic relatedness of the expanded tweets to word lists representing positive and negative sentiment. This approach helps to assess the polarity of tweets that do not directly contain polarity cues. Moreover, we incorporate syntactic, lexical and surface sentiment features. On the message level, our system achieved the 8th place in terms of macro-averaged F-score among 50 systems, with particularly good performance on the Life-Journal corpus (F_1=71.92) and the Twitter sarcasm (F_1=54.59) dataset. On the expression level, our system ranked 14 out of 27 systems, based on macro-averaged F-score.

1 Introduction

Microblogging sites, such as Twitter, have become an important source of information about current events. The fact that users write about their experiences, often directly during or shortly after an event, contributes to the high level of emotions in many such messages. Being able to automatically and reliably evaluate these emotions in context of a specific event or a product would be highly beneficial not only in marketing (Jansen et al., 2009) or public relations, but also in political sciences (O'Connor et al., 2010), disaster management, stock market analysis (Bollen et al., 2011) or the health sector (Culotta, 2010).

Due to its large number of applications, sentiment analysis on Twitter is a very popular task. Challenges arise both from the character of the task and from the language specifics of Twitter messages. Messages are normally very short and informal, frequently using slang, alternative spelling, neologism and links, and mostly ignoring the punctuation.

Our experiments have been carried out as part of the SemEval 2014 Task 9 - Sentiment Analysis on Twitter (Rosenthal et al., 2014), a rerun of a SemEval-2013 Task 2 (Nakov et al., 2013). The datasets are thus described in detail in the overview papers. The rerun uses the same training and development data, but new test data from Twitter and a "surprise domain". The task consists of two subtasks: an expression-level subtask (Subtask A) and a message-level subtask (Subtask B). In subtask A, each tweet in a corpus contained a marked instance of a word or phrase. The goal is to determine whether that instance is positive, negative or neutral in that context. In subtask B, the goal is to classify whether the entire message is of positive, negative, or neutral sentiment. For messages conveying both a positive and negative sentiment, the stronger one should be chosen.

The key components of our system are the sentiment polarity lexicons. In contrast to previous approaches, we do not only count exact lexicon hits, but also calculate explicit semantic relatedness (Gabrilovich and Markovitch, 2007) between the tweet and the sentiment list, benefiting from resources such as Wiktionary and WordNet. On top of that, we expand content words (adjectives, adverbs, nouns and verbs) in the tweet with similar words, which we derive from a novel corpus of over 80 million English Tweets gathered by the Language Technology group[1] at TU Darmstadt.

[1] http://www.lt.informatik.tu-darmstadt.de

2 Experimental setup

Our experimental setup is based on an open-source text classification framework *DKPro TC*[2] (Daxenberger et al., 2014), which allows to combine NLP pipelines into a configurable and modular system for preprocessing, feature extraction and classification. We use the *unit classification* mode of DKPro TC for Subtask A and the *document classification* mode for Subtask B.

2.1 Preprocessing

We customized the message reader for Subtask B to ignore the first part of the tweet when the word *but* is found. This approach helps to reduce the misleading positive hits when a negative message is introduced positively (*It'd be good, but*).

For preprocessing the data, we use components from DKPro Core[3]. Preprocessing is the same for subtasks A and B, with the only difference that in the subtask A the target expression is additionally annotated as *text classification unit*, while the rest of the tweet is considered to be a document context. We first segment the data with the Stanford Segmenter[4], apply the Stanford POS Tagger with a Twitter-trained model (Derczynski et al., 2013), and subsequently apply the Stanford Lemmatizer[4], TreeTagger Chunker (Schmid, 1994), Stanford Named Entity Recognizer (Finkel et al., 2005) and Stanford Parser (Klein and Manning, 2003) to each tweet. After this linguistic preprocessing, the token segmentation of the Stanford tools is removed and overwritten by the ArkTweet Tagger (Gimpel et al., 2011), which is more suitable for recognizing hashtags and smileys as one particular token. Finally, we expand the tweet and proceed to feature extraction as described in detail in Section 3.

2.2 Classification

We trained our system on the provided training data only, excluding the `dev` data. We use classifiers from the WEKA (Hall et al., 2009) toolkit, which are integrated in the DKPro TC framework. Our final configuration consists of a SVM-SMO classifier with a gaussian kernel. The optimal hyperparameters have been experimentally derived and finally set to C=1 and G=0.01. The resulting model was wrapped in a cost sensitive meta classifier from the WEKA toolkit with the error costs set to reflect the class imbalance in the training set.

3 Features used

We now describe the features used in our experiments. For Subtask A (contextual polarity), we extracted each feature twice - once on the tweet level and once on the focus expression level. Only n-gram features were extracted solely from the expressions. For Subtask B (tweet polarity), we extracted features on tweet level only. In both cases, we use the Information Gain feature selection approach in WEKA to rank the features and prune the feature space with a threshold of T=0.005.

3.1 Lexical features

As a basis for our similarity and expansion experiments (sections 3.4 and 3.5), we use the binary sentiment polarity lexicon by Liu (2012) augmented with the smiley polarity lexicon by Becker et al. (2013) and an additional swear word list[5] [further as $Liu_{augmented}$]. We selected this augmented lexicon for two reasons: firstly, it was the highest ranked lexical feature on the development-test and crossvalidation experiments, secondly it consists of two plain word lists and therefore does not introduce another complexity dimension for advanced feature calculations.

We further measure lexicon hits normalized per number of tweet tokens for the following lexicons: Pennebaker's Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2001), the NRC Emotion Lexicon (Mohammad and Turney, 2013), the NRC Hashtag Emotion Lexicon (Mohammad et al., 2013) and the Sentiment140 lexicon (Mohammad et al., 2013). We use an additional lexicon of positive, negative, very positive and very negative words, diminishers, intensifiers and negations composed by Steinberger et al. (2012), where we calculate the polarity score as described in their paper.

In a complementary set of features we combine each of the lexicons above with a list of weighted intensifying expressions from Taboada et al. (2011). The intensity of any polar word found in any of the emotion lexicons used is intensified or diminished by a given weight if an intensifier (*a bit, very, slightly...*) is found within the preceding three tokens.

[2] http://code.google.com/p/dkpro-tc

[3] http://code.google.com/p/dkpro-core-asl

[4] http://nlp.stanford.edu/software/corenlp.shtml

[5] based on http://www.youswear.com

Additionally, we record the overall counts of lexicon hits for positive words, negative words and the difference of the two. In one set of features we consider only lexicons clearly meant for binary polarity, while a second set of features also includes other emotions, such as fear or anger, from the NRC and the LIWC corpora.

3.2 Negation

We handle negation in two ways. On the expression level (Subtask A) we rely on the negation dependency tag provided by the Stanford Dependency Parser. This one captures verb negations rather precisely and thus helps to handle emotional verb expressions such as *like* vs *don't like*. On the tweet level (all features of Subtask B and entire-tweet-level features of Subtask A) we adopt the approach of Pang et al. (2002), considering as a negation context any sequence of tokens between a negation expression and the end of a sentence segment as annotated by the Stanford Segmenter. The negation expressions *(don't, can't...)* are represented by the list of invertors from Steinberger's lexicon (Steinberger et al., 2012). We first assign polarity score to each word in the tweet based on the lexicon hits and then revert it for the words lying in the negation context. This approach is more robust than the one of the dependency governor but is error-prone in the area of overlapping (cascaded) negation contexts.

3.3 N-gram features

We extract the 5,000 most frequent word unigrams, bigrams and trigrams cleaned with the Snowball stopword list[6] as well as the same amount of skip-n-grams and character trigrams. These are extracted separately on the target expression level for subtask A and on document level for subtask B. On the syntactic level, we monitor the most frequent 5,000 part-of-speech ngrams with the size up to part-of-speech quadruples. Additionally, as an approximation for exploiting the key message of the sentence, we extract from the tweets a verb chunk and its left and right neighboring noun chunks, obtaining combinations such as *we-go-cinema*. The 1,000 most frequent chunk triples are then used as features similarly to ngrams.

Word	Score	Word (continued)	Score
awesome	1,000	fun	60
amazing	194	sexy	59
great	148	cold	59
cool	104	crazy	57
good	96	fantastic	56
best	93	bored	55
beautiful	93	excited	54
nice	87	true	53
funny	84	stupid	53
cute	81	gr8	52
perfect	70	entertaining	52
wonderful	67	favorite	52
lovely	66	talented	49
tired	65	other	49
annoying	63	depressing	48
Great	63	flawless	48
new	62	inspiring	47
hilarious	62	incredible	46
bad	61	complicated	46
hot	61	gorgeous	45

Table 1: Unsupervised expansion of '*awesome*'

3.4 Tweet expansion

We expanded the content words in a tweet, i.e. nouns, verbs, adjectives and adverbs, with similar words from a word similarity thesaurus that was computed on 80 million English tweets from 2012 using the JoBim contextual semantics framework (Biemann and Riedl, 2013). Table 1 shows an example for a lexical expansion of the word *awesome*. The score was computed using left and right neighbor bigram features for the holing operation. The value hence shows how often the word appeared in the same left and right context as the original word. The upper limit of the score is set to 1,000.

We then match the expanded tweet against the $Liu_{augmented}$ positive and negative lexicons. We assign to the lexicon hits of the expanded words their (contextual similarity) expansion score, using a score of 1,000 as an anchor-value for the original tweet, setting an expansion cut at 100. The overall tweet score is then normalized by the sum of word expansion scores.

3.5 Semantic similarity

Tweet messages are short and each emotional word is very valuable for the task, even when it may not be present in a specific lexicon. Therefore, we calculate a semantic relatedness score between the tweet and the positive or negative word list. We use the ESA similarity measure

[6] http://snowball.tartarus.org/algorithms/english/stop.txt

(Gabrilovich and Markovitch, 2007) as implemented in the DKPro similarity software package (Bär et al., 2013), calculated on English Wiktionary and WordNet as two separate concept spaces. The ESA vectors are freely available[7]. This way we obtain in total six features: *sim(original tweet word list, positive word list), sim(original tweet word list, negative word list)*, difference between the two, *sim(expanded tweet word list, positive word list), sim(expanded tweet word list, negative word list)* and difference between the two. Our SemEval run was submitted using WordNet vectors mainly for the shorter computation time and lower memory requirements. However, in our later experiments Wiktionary performed better. We presume this can be due to a better coverage for the Twitter corpus, although detailed analysis of this aspect is yet to be performed.

3.6 Other features

Pak and Paroubek (2010) pointed out a relation between the presence of different part-of-speech types and sentiment polarity. We measure the ratio of each part-of-speech type to each chunk. We furthermore count the occurrences of the dependency tag for negation. We use the Stanford Named Entity Recognizer to count occurrence of persons, organizations and locations in the tweet. Additionaly, beside basic surface metrics, such as the number of tokens, characters and sentences, we measure the number of elongated words (such as *coool*) in a tweet, ratio of sentences ending with exclamation, ratio of questions and number of positive and negative smileys and their proportion. We capture the smileys with the following two regular expressions for positive, respectively negative ones: `[<>]?[:;=8][-o*']?[)]dDpPxXoO0*}]`, `[<>]?[:;=8][-o*']?[([/:{|]`. We also separately measure the sentiment of smileys at the end of the tweet body, i.e. followed only by a hashtag, hyperlink or nothing.

4 Results

In Subtask A, our system achieved an averaged F-score of 81.42 on the LiveJournal corpus and 79.67 on the Twitter 2014 corpus. The highest scores achieved in related work were 85.61 and 86.63 respectively. For subtask B, we scored 71.92 on LifeJournal and 63.77 on Twitter 2014, while the highest F-scores reported by related work were 74.84 and 70.96.

Features with the highest Information Gain were the ones based on $Liu_{augmented}$. They were followed by features derived from the lexicons of Steinberger, which includes invertors, intensifiers and four polarity levels of words. On the other hand, adding the weighted intensifiers of Taboada et al. (2011) to lexicons did not outperform the simple lexicon lookup. Overall, lexicon-based features contributed to the highest performance gain, as shown in Table 3. The negation approach based on the Stanford dependency parser was the most helpful, although it tripled the runtime. Using the simpler negation context as suggested in Pang et al. (2002) performed still on average better than using none.

When using WordNet, semantic similarity to lexicons did not outperform direct lexicon hits. Usage of Wiktionary instead lead to major improvement (Table 3), unfortunately after the SemEval challenge.

Tweet expansion appears to improve the classification performance, however the threshold of 100 that we used in our setup was chosed too conservatively, expanding mainly stopwords with other stopwords or words with their spelling alternatives, resulting in a noisy, little valuable feature (`expansion full` in Table 3). Setting up the threshold to 50 and cleaning up both the tweet and the expansion with Snowball stopword list (`expansion clean` in Table 3), the performance increased remarkably.

Amongst other prominent features were parts of lexicons such as LIWC Positive emotions, LIWC Affect, LIWC Negative emotions, NRC Joy, NRC Anger and NRC Disgust. Informative were also the proportions of nouns, verbs and adverbs, the exclamation ratio or number of positive and negative smileys at the end of the tweet.

5 Error analysis

Table 2 lists a sample of misclassified messages. The majority of errors resulted from misclassifying neutral tweets as emotionally charged. This was partly caused by the usage of emoticons and expressions such as *haha* in a neutral context, such as in examples 2 and 3. Other errors were caused by lexicon hits of proper nouns (example 1), or by using negative words and swearwords in an

[7] https://code.google.com/p/dkpro-similarity-asl/downloads/list

#	Gold label	Prediction	Message
1	negative	positive	Your plans of attending the **Great** Yorkshire Show may have been washed out because of the weather, so how about...
2	neutral	positive	sitting here with my belt in jean shorts watching Cena win his first title. I think we tie for 1st my friend **xD**
3	neutral	positive	saw your LJ post ... yay for Aussies **;)**
4	positive	negative	haha , that **sucks** , because the drumline will be just fine
5	positive	negative	...woah, Deezer. Babel only came out on Monday, can you leave it up for longer than a day to give slow people like me a chance?
6	positive	negative	Yeah so much has changed for the 6th. Lots of combat **fighting**. And inventory is different.
7	positive	negative	just finish doing it and tomorrow I'm going to the celtics game and don't **fucking** say ”thanks for the invite” it's **annoying**
8	positive	negative	Haha... Yup hopefully we will **lose** a few kg by mon. after hip hop can go orchard and weigh
9	positive	negative	U r just like my friends? I made them feel warm, happy, then make them angry and they cry? Finally they left me? Will u leave 2? I hope not. Really hope so.

Table 2: Examples of misclassified messages

overall positive tweet (examples 4, 7, 9). Some of the tweets contained domain specific vocabulary that would hit the negative lexicon, e.g., discussing fighting and violence in computer games would, in contrast to other topic domains, usually have positive polarity (example 6). Similar domain-specific polarity distinction could be applied to certain verbs, e.g., *lose weight* vs. *lose a game* (example 8).

Another challenge for the system was the non-standard language in twitter with a large number of spelling variants, which was only partly captured by the emotion lexicons tailored for this domain. A twitter-specific lemmatizer, which would group all variations of a misspelled word into one, could help to improve the performance.

The length of the negation context window does not suit all purposes. Also double negations such as *I don't think he couldn't...* can easily misdirect the polarity score.

6 Conclusion

We presented a sentiment classification system that can be used on both message level and expression level with only small changes in the framework configuration. We employed a contextual similarity thesaurus for the lexical expansion of the messages. The expansion was not efficient without an extensive stopword cleaning, overweighting more common words and introducing noise. Utilizing the semantic similarity of tweets to lexicons instead of a direct match improves the score only with certain lexicons, possibly dependent on the coverage. Negation by dependency parsing was more beneficial to the classifier than the negation by keyword span annotation. Naive combination of sentiment lexicons was not more helpful than using individual ones separately. Among the common sources of errors were laughing signs used in neutral messages and swearing used in positive messages. Even within Twitter, the same words can have different polarity in different domains (*lose weight, lose game, game with nice violent fights...*). Deeper semantic insights are necessary to distinguish between polar words in context.

Feature(s)	ΔF_1 **Twitter2014**	ΔF_1 **LifeJournal**
Similarity Wikt.	0.56	3.65
Similarity WN	0.0	2.61
Expansion full	0.0	0.0
Expansion clean	0.59	3.82
Lexical negation	0.24	0.13
N-gram features	0.30	0.32
Lexicon-based f.	7.85	4.74

Table 3: Performance increase where feature added to the full setup

7 Acknowledgement

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806. We warmly thank Chris Biemann, Martin Riedl and Eugen Ruppert of the Language Technology group at TU Darmstadt for providing us with the Twitter-based distributional thesaurus.

References

Daniel Bär, Torsten Zesch, and Iryna Gurevych. 2013. Dkpro similarity: An open source framework for text similarity. In *Proceedings of the 51st Annual*

Meeting of the Association for Computational Linguistics: System Demonstrations, pages 121–126, Sofia, Bulgaria.

Lee Becker, George Erhart, David Skiba, and Valentine Matula. 2013. Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. *Atlanta, Georgia, USA*, page 333.

Chris Biemann and Martin Riedl. 2013. Text: Now in 2d! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8.

Aron Culotta. 2010. Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the First Workshop on Social Media Analytics*, pages 115–122, New York, NY, USA.

Johannes Daxenberger, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch. 2014. Dkpro tc: A java-based framework for supervised learning experiments on textual data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, page (to appear), Baltimore, MD, USA.

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*, Hissar, Bulgaria.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, volume 7, pages 1606–1611, Hyderabad, India.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Bernard J. Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 60(11):2169–2188.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430.

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

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

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.

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.

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 *Fourth International AAAI Conference on Weblogs and Social Media*, pages 122–129.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

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.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In Preslav Nakov and

Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, volume 12, pages 44–49.

Josef Steinberger, Mohamed Ebrahim, Maud Ehrmann, Ali Hurriyetoglu, Mijail Kabadjov, Polina Lenkova, Ralf Steinberger, Hristo Tanev, Silvia Vázquez, and Vanni Zavarella. 2012. Creating sentiment dictionaries via triangulation. *Decision Support Systems*, 53(4):689–694.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.