The Impact of Figurative Language on Sentiment Analysis

Tomáš Hercig[†‡]
[‡] NTIS – New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Technická 8, 306 14 Plzeň Czech Republic
tigi@kiv.zcu.cz

Ladislav Lenc[†‡]
[†] Department of Computer Science and Engineering, Faculty of Applied Sciences University of West Bohemia, Univerzitní 8, 306 14 Plzeň Czech Republic
llenc@kiv.zcu.cz

Abstract

Figurative language such as irony, sarcasm, and metaphor is considered a significant challenge in sentiment analysis. These figurative devices can sculpt the affect of an utterance and test the limits of sentiment analysis of supposedly literal texts. We explore the effect of figurative language on sentiment analysis. We incorporate the figurative language indicators into the sentiment analysis process and compare the results with and without the additional information about them. We evaluate on the SemEval-2015 Task 11 data and outperform the first team with our convolutional neural network model and additional training data in terms of mean squared error and we follow closely behind the first place in terms of cosine similarity.

1 Introduction

Recently there have been several experiments with sarcasm detection e.g. (Ptáček et al., 2014; Ghosh and Veale, 2016; Zhang et al., 2016; Poria et al., 2016). Although these works succeeded in their goal to detect variations of sarcasm, one final step is still missing – the evaluation of sentiment analysis with and without additional sarcasm indicators. There have been attempts at investigating the impact of sarcasm on sentiment analysis (Maynard and Greenwood, 2014) or thorough analysis of hashtags indicating sarcastic tweets (Sulis et al., 2016). However, the impact of figurative language (including sarcasm) on sentiment analysis has not yet been studied in depth.

Our goal is to explore the effect of figurative language (e.g. sarcasm) on sentiment analysis. We incorporate the figurative language indicators into the sentiment analysis process and compare the results with and without the additional information about them. We use additional dataset to examine the results achieved with extra training data.

2 Related Work

2.1 Sarcasm Detection

Maynard and Greenwood (2014) performed experiments with a rule based approach to sarcasm detection and sentiment analysis. They manually annotated 266 sentences from 134 collected tweets. Their corpus contains 68 opinionated sentences (62 negative, 6 positive), out of these 61 were deemed to be sarcastic. Their regular sentiment polarity analyser achieved 0.27 accuracy while the sentiment polarity analyser considering sarcasm achieved 0.77 accuracy using handcrafted rules and lexicons. However this dataset is imbalanced and very small to draw any conclusions.

Second experiment measured the accuracy of sarcasm and polarity detection. The corpus consists of 400 tweets (91 sarcastic sentences). Regrettably, the previous regular vs. sarcasm analyser comparison exploring the impact of sarcasm on polarity detection is not included. They only measured the performance of the sarcastic analyser.

The detection of sarcasm in Czech and English was done by Ptáček et al. (2014). They created large Czech Twitter corpus consisting of 7k manually-labeled tweets and provide it to the community along with the automatically-labeled English balanced (50k sarcastic, 50k normal tweets) and imbalanced corpora (25k sarcastic, 75k normal tweets). They evaluated two classifiers with various combinations of features on each dataset achieving F1 score of 0.947 and 0.924 on the balanced and imbalanced datasets, respectively and

Proceedings of Recent Advances in Natural Language Processing, pages 301–308,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_041

F1 score 0.582 on the Czech imbalanced dataset.

2.2 SemEval Workshop

SemEval-2015 Task 10B (Rosenthal et al., 2015) Sentiment analysis in Twitter is a re-run of previous years (SemEval-2013 Task 2 and SemEval-2014 Task 9). The goal of this task is to classify Twitter messages (tweets) into positive, negative, or neutral sentiment classes. Teams evaluate their results on five datasets from previous years and on two new datasets.

Astudillo et al. (2015) treat sentiment analysis as a regression problem which allows more fine-grained sentiment assessment. They model tweets using word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) embeddings that are averaged or summed over the given tweet. A regression model is than trained on the resulting representations. This model achieved the fourth place.

The goal of **SemEval-2015 Task 11** was to perform fine-grained sentiment analysis over texts containing figurative language. Ghosh et al. (2015) have created a dataset of figurative tweets using Twitter4j API and a set of hashtag queries (*#sarcasm*, *#sarcastic*, *#irony* and words such as *figuratively*). The dataset has been annotated for sentiment analysis on a fine-grained 11-point scale (-5 to 5, including 0). Evaluation measures for this task were mean squared error (MSE) and cosine similarity, both with penalization for not giving scores for all tweets.

CLaC (Özdemir and Bergler, 2015b) presented the best result for SemEval-2015 Task 11 using decision tree regression M5P (Wang and Witten, 1997). They combined various lexicons with negation and modality scopes. They also participated in SemEval-2015 Task 10B achieving ninth place. Özdemir and Bergler (2015a) performed a comprehensive ablation study of features.

CPH (McGillion et al., 2015)[1] and PRHLT (Gupta and Gómez, 2015)[2] teams did not use lexicons and therefore provide comparable baselines to our models, which also do not use lexicons.

Sulis et al. (2016) analyse the corpus from Semeval-2015 Task 11 in terms of hashtags (*#irony*, *#sarcasm*, and *#not*) and confirm that messages using figurative language mostly express a negative sentiment. They experimented with binary classification (separation) of tweets with these hashtags.

[1]They used ensemble methods and ridge regression.

[2]They used ensembles of extremely random trees with character n-grams.

2.3 Neural Networks

A Convolutional Neural Network (CNN) architecture for sentence classification is proposed in (Kim, 2014). The network uses several convolutional kernel sizes simultaneously. It utilizes pre-trained word2vec embeddings. The approach was tested on several tasks including sentiment analysis. It proved state-of-the-art results on the Stanford sentiment treebank (Socher et al., 2013) both for binary and five-level sentiment classification. Best results are obtained using pre-trained embeddings for initialization of the embedding layer.

A neural network model for sarcasm detection is proposed in (Ghosh and Veale, 2016). The model is composed from a CNN followed by a long short term memory (LSTM) network. First a CNN is applied to the input. LSTM is then applied directly on the output of the convolutional layer. Output of the LSTM is fed to a fully connected layer and a softmax layer determines the class. F-score of 0.92 is achieved on their dataset containing 39k tweets.

Another approach is presented in (Zhang et al., 2016). A deep neural network is used for tweet sarcasm detection. The network has two components for local and contextual (history) tweets. The local one is a bi-directional gated recurrent unit that extracts dense real-valued output. The other component applies a pooling layer directly to the word embeddings for words in the contextual tweets and maps it to a fixed length vector. A hidden layer then combines these two components and is followed by a softmax layer. Embeddings are initialized using GloVe. Results are compared with manually created features.

CNNs are utilized for feature extraction in (Poria et al., 2016). Sentiment, emotion, and personality features are utilized for sarcasm detection. CNN models are separately trained on datasets corresponding to the three types of features. The three CNNs are then merged. The final classification is done either using a support vector machines classifier or another CNN which uses the merged features as a static channel and connects it to the penultimate layer before the softmax layer.

Type	Train		Test		Trial	
	Mean Polarity	# Tweets	Mean Polarity	# Tweets	Mean Polarity	# Tweets
Sarcasm	-2.25	5000	-2.02	1200	-1.94	746
Irony	-1.70	1000	-1.87	800	-1.35	81
Metaphor	-1.49	2000	-0.77	800	-0.34	198
Other	–	–	-0.26	1200	–	–
Overall	-1.99	8000	-1.21	4000	-1.89	1025

Table 1: The tweet distributions and mean polarity in SemEval-2015 Task 11 datasets.

Type	Train		Test		Trial	
	Mean Polarity	# Tweets	Mean Polarity	# Tweets	Mean Polarity	# Tweets
Sarcasm	-2.25	4895	-2.05	1107	-2.00	612
Irony	-1.70	1424	-1.85	763	-1.98	23
Metaphor	-1.49	1681	-0.85	878	-0.67	91
Other	–	–	-0.33	1252	–	–
Overall	-1.99	8000	-1.21	4000	-1.83	726

Table 2: The tweet distributions and mean polarity in SemEval-2015 Task 11 datasets by hashtags.

3 Datasets

We use the dataset from SemEval-2015 Task 11 (Ghosh et al., 2015) for training and evaluation. Table 1 shows the mean polarity and the original estimated tweet distributions[3]. The category type labels refer to the authors' expectations of tweet category types in each segment of the dataset. To ensure the validity of the task, the authors added the category *other* to the test dataset.

Table 2 contains the same statistics for our collected datasets[4]. We separated data into the category types by using the harvesting criteria for the datasets' collection (e.g. the *#irony* hashtag)[5]. Table 3 shows the detailed sentiment polarity distributions. The training data were provided with rounded integer values and floating point values. However when we rounded the real-valued scores we got different counts for individual polarity values. This issue corresponds to the *Train* data columns *int* and *rounded*. In our experiments we use *rounded* values wherever it is possible.

To compensate for the missing *other* category in the training data of SemEval-2015 Task 11, we use the dataset from SemEval-2015 Task 10B (Rosenthal et al., 2015) as additional training data. We were able to download approximately 75.7% of the training data and 78.6% of the test data (see Table 4).

For the SemEval-2015 Task 10B we evaluate on the test data and the sarcasm dataset[6] from the same task in SemEval-2014.

4 Convolutional Neural Network

The architecture of the proposed CNN is depicted in Figure 1. We use similar architecture to the one proposed by Lenc and Král (2017). The input layer of the network receives a sequence of word indices from a dictionary. The input vector must be of a fixed length. We solve this issue by padding the input sequence to the maximum tweet length denoted M. A special "PADDING" token is used for this purpose. The embedding layer maps the word indices to the real-valued embedding vectors of length L. The convolutional layer consists of N_C kernels containing $k \times 1$ units and uses rectified linear unit (ReLU) activation function. The convolutioanl layer is followed by a max-pooling layer and dropout for regularization. The max-pooling layer takes maxima from patches with dimensions $(M - k + 1) \times 1$. The output of the max-pooling layer is fed into a fully-connected layer. The fully connected layer is optionally concatenated with the additional `category-type-binary-input`

[3]In the original publication there were some typos, we show the recalculated statistics.

[4]Note that we were unable to download the whole Trial dataset due to perishability of tweets.

[5]Separating tweets into category types is a rule based approach.

[6]We were not able to download sufficient amount of tweets for the sarcasm dataset from SemEval-2015 Task 10B.

Value	Test	Train (int)	Train (rounded)	Trial orig.	Trial downl.
-5	4	0	6	6	4
-4	100	361	364	90	56
-3	737	2954	2971	403	282
-2	1541	2911	2934	255	180
-1	680	909	861	87	67
0	298	347	345	50	40
1	169	164	165	51	39
2	155	197	197	41	29
3	201	106	106	32	23
4	111	49	49	9	6
5	4	2	2	1	0
SUM	4000	8000	8000	1025	726

Table 3: The tweet sentiment polarity distributions in SemEval-2015 Task 11.

Corpus	Positive	Negative	Neutral	Total	Downloaded
Twitter2015-train	3,640	1,458	4,586	9,684	7,326 (76%)
Twitter2015-test	1,038	365	987	2,390	1,878 (79%)
Twitter2014-sarcasm	33	40	13	86	86 (100%)

Table 4: The tweet polarity distributions in SemEval-2015 Task 10B.

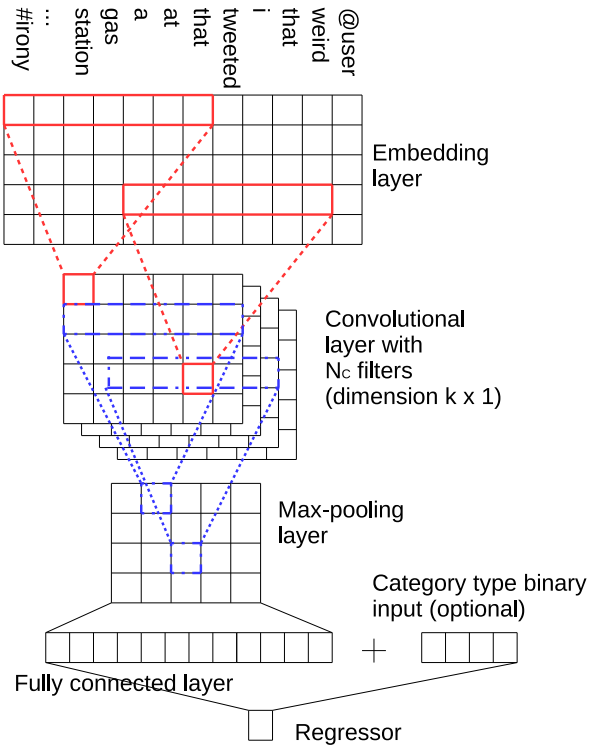

Figure 1: Neural network architecture.

layer that adds the information about hashtags used in the tweet. The output layer is connected to this layer and has just one neuron serving as a regressor.

In our experimental setup we use the embedding dimension $L = 300$ and $N_C = 40$ convolutional kernels with 5×1 units. The penultimate fully-connected layer contains 256 neurons connected with the optional `category-type-binary input` with 4 neurons. We train the network using adaptive moment estimation optimization algorithm (Kingma and Ba, 2014). Mean square error is used as loss function.

5 Experiments

We perform regression experiments on the 11-point scale (-5, .., 0, .., 5) for the SemEval-2015 task 11 and classification into positive, negative, and neutral classes for SemEval-2015 task 10B.

5.1 Preprocessing

The same preprocessing has been done for all datasets. We use UDPipe (Straka et al., 2016) with English Universal Dependencies 1.2 models for POS tagging and lemmatization. Tokenization has been done by TweetNLP tokenizer (Owoputi et al., 2013). Preliminary experiments have shown that lower-casing the data achieves slightly better re-

sults, thus all the experiments are performed with lower-cased data. We further replace all user mentions with the token "@USER" and all links with the token "$LINK".

5.2 Regression

Regression has been done using CNN (Section 4) and Weka 3.6.6 (Hall et al., 2009) with the M5P decision tree regression. We use the SemEval-2015 task 11 scorer to evaluate our results. Used features are unigrams with more than two occurrences. We map the additional training data from SemEval-2015 task 10B (-1 negative, 0 neutral, 1 positive) to the 11-point scale by using multiplier 4 (-4 negative, 0 neutral, 4 positive). This corresponds to our intuition that the positive and negative class should contain strong polarity values.

We incorporate the figurative language indicators into the sentiment analysis process and compare the results with and without the additional information about them. We use additional dataset to examine the results achieved with extra training data and to compensate for the missing *other* category in the training data.

First we use the preprocessed dataset. Then we remove the category types harvesting criteria (e.g. the #irony hashtag) from the entire dataset. Finally we add binary features indicating category types to the second experiment.

Table 5 shows the regression results, where system description "`-nohash`" indicates removing the category types and "`-nohash + #`" signifies the same plus binary features indicating category types.[7]

Removing the category type indicators deteriorates the results for most cases, except for the category *Metaphor* without additional training data, where the results are actually better. We believe this is due to the removal of words that results in less uncertainty for the model. A similar case is the CNN model for *Irony* without additional training data.

Restoration of the category types using binary features again improves the results in most cases, with the exception of the category *Metaphor*. This suggests that figurative language does matter and information about the given figurative language helps improve sentiment analysis.

Metaphor seems to be very hard to correctly assign sentiment polarity. We believe this is caused by the datasets' composition, because the training dataset does not contain the category *Other*, thus the tweets that do not belong into the *Irony* or *Sarcasm* categories must belong to the *Metaphor* category. This claim presumes that the *Other* category is not present in the training dataset. We believe this is the reason why the *Metaphor* category is suffering in the "`-nohash + #`" setting. Moreover, tweets from training data in this category such as "@USER we're the proverbial frog getting slowly boiled in the pot of water." may not contain words that can be removed as figurative language indicators.

Additional training data directly improves results for *Metaphor* and *Other*, however the results for *Sarcasm* and *Irony* are worse. This effect is diminished in the "`-nohash + #`" setting. The results for the "`-nohash`" setting are consistently worse for all category types.

The best results are achieved with additional training data and basic setting with best results for the category types *Metaphor* and *Other*, which confirms the claim by Ghosh et al. (2015) i.e. there is a strong correlation between the overall performance and performance on the category *Metaphor* and *Other*.

Regardless of the categories, the *Overall* column in Table 5 is directly comparable to the SemEval-2015 Task 11 results. We can see that removing the figurative language indicators always deteriorates the results and their restoration by the binary figurative language features again improves the results for all cases. This supports our hypothesis that figurative language affects sentiment analysis.

5.3 Classification

The classification experiment in Table 6 was performed using the maximum entropy classifier (MaxEnt) from Brainy (Konkol, 2014). This experiment shows that even small in-domain (sarcasm) training data can help improve results. Used features are unigrams and bigrams with more than five occurrences. We train the maximum entropy classifier on the SemEval-2015 Task 10B training data (Twitter2013-train cleansed) and test on Twitter2015-test data and the Twitter2014-sarcasm data.

The F1 score for test data changes just slightly with additional training data (tweets containing

[7] Note that the category results are not directly comparable to the SemEval-2015 task 11 results.

Train Data	System Description	Sarcasm		Irony		Metaphor		Other		Overall	
		Cosine	MSE	Cosine	MSE	Cosine	MSE	Cosine	MSE	Cosine	MSE
T11	Best	**0.904**	**0.934**	**0.918**	**0.673**	**0.655**	**3.155**	**0.612**	**3.411**	**0.758**	**2.117**
T11	CLaC	0.892	1.023	0.904	0.779	**0.655**	**3.155**	0.584	**3.411**	**0.758**	**2.117**
T11	CPH	0.897	0.971	0.886	0.774	0.325	5.014	0.218	5.429	0.625	3.078
T11	PRHLT	0.891	1.028	0.901	0.784	0.167	5.446	0.218	4.888	0.623	3.023
T11	CNN	**0.908**	**0.893**	0.863	1.049	0.402	4.641	0.361	4.408	0.652	2.846
T11	-nohash	0.901	0.942	**0.886**	**0.897**	0.420	4.554	0.236	5.822	0.606	3.254
T11	-nohash + #	0.899	0.995	0.879	0.928	0.277	5.134	0.291	4.772	0.620	3.073
T10+T11	CNN	0.900	0.957	0.880	0.924	**0.620**	**3.401**	**0.633**	**2.966**	**0.755**	**2.116**
T10+T11	-nohash	0.851	1.523	0.860	1.163	0.547	3.876	0.518	3.786	0.691	2.679
T10+T11	-nohash + #	0.880	1.269	0.876	0.976	0.573	3.759	0.591	3.219	0.724	2.370
T11	M5P	0.908	0.888	**0.903**	**0.802**	0.291	5.040	0.277	4.588	0.636	2.941
T11	-nohash	0.910	0.874	0.876	0.962	0.378	4.921	0.190	4.917	0.625	3.045
T11	-nohash + #	0.909	0.893	0.891	0.845	0.357	4.825	0.274	4.599	0.640	2.907
T10+T11	M5P	0.834	1.720	0.863	1.140	**0.525**	**3.986**	**0.410**	**4.121**	**0.658**	**2.858**
T10+T11	-nohash	0.816	1.678	0.832	1.295	0.468	4.341	0.388	4.469	0.623	3.063
T10+T11	-nohash + #	**0.912**	**0.858**	0.877	0.958	0.397	4.639	0.381	4.549	0.654	2.862

Table 5: Results on the SemEval-2015 Task 11. Training data T11 and T10 denote the respective tasks' datasets used for training. System description "`-nohash`" indicates removing the category types harvesting criteria (e.g. the #irony hashtag), "`-nohash + #`" signifies the same plus binary features indicating category types.

sarcasm from SemEval-2015 Task 11 trial data[8]). The additional training data cause slight improvement on the test data and greatly improve the results on the sarcasm dataset. We would have achieved the seventh place out of 40 participants on the sarcasm dataset. Our simple solution is competitive on the sarcasm dataset with the best results achieved with lexicons, classifier ensembles, and various dictionaries.

Description	Test F1	Sarcasm F1
Best Result	0.648	0.591
CLaC	0.620	0.514
MaxEnt	0.527	0.457
MaxEnt + trial	0.533	0.547

Table 6: Results on the SemEval-2015 Task 10B.

6 Conclusion

In this article we have shown that figurative language can affect sentiment analysis. In our regression experiments removing the figurative language indicators deteriorates the results and their restoration by the binary figurative language features again improves the results on the whole dataset. The classification experiment shows that even small in-domain (sarcasm) training data can help improve results.

Our approach is simple without fine-tuned features and lexicons. We only use extra training data, which was allowed for this task. In the SemEval-2015 Task 11 we would have ranked first with CNN and additional training data in terms of MSE and second in terms of Cosine similarity. Our CNN model without additional training data would have achieved the fourth place in terms of MSE and the seventh place in terms of Cosine similarity.

In the future, we plan to create a dataset with explicitly marked categories for figurative language in both training and test data. Then we will repeat all experiments on this new dataset and compare the results.

Acknowledgments

This work was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports and by Grant No. SGS-2016-018 Data and Software Engineering for Advanced Applications.

[8] We mark tweets as positive for polarity ≥ 1 and negative for polarity ≤ -1.

References

Ramón Astudillo, Silvio Amir, Wang Ling, Bruno Martins, Mario J. Silva, and Isabel Trancoso. 2015. INESC-ID: Sentiment Analysis without Hand-Coded Features or Linguistic Resources using Embedding Subspaces. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 652–656. http://www.aclweb.org/anthology/S15-2109.

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. 2015. SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 470–478. http://www.aclweb.org/anthology/S15-2080.

Aniruddha Ghosh and Dr. Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, San Diego, California, pages 161–169. http://www.aclweb.org/anthology/W16-0425.

Parth Gupta and Jon Ander Gómez. 2015. PRHLT: Combination of Deep Autoencoders with Classification and Regression Techniques for SemEval-2015 Task 11. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 689–693. http://www.aclweb.org/anthology/S15-2116.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations* 11(1):10–18. http://www.sigkdd.org/explorations/issues/11-1-2009-07/p2V11n1.pdf.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1746–1751. http://www.aclweb.org/anthology/D14-1181.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Michal Konkol. 2014. Brainy: A machine learning library. In Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi Zadeh, and Jacek Zurada, editors, *Artificial Intelligence and Soft Computing*, Springer International Publishing, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499.

Ladislav Lenc and Pavel Král. 2017. Deep neural networks for czech multi-label document classification. *CoRR* abs/1701.03849. http://arxiv.org/abs/1701.03849.

Diana Maynard and Mark Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

Sarah McGillion, Héctor Martínez Alonso, and Barbara Plank. 2015. CPH: Sentiment analysis of Figurative Language on Twitter #easypeasy #not. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 699–703. http://www.aclweb.org/anthology/S15-2118.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781. http://arxiv.org/abs/1301.3781.

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 the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, pages 380–390. http://www.aclweb.org/anthology/N13-1039.

Canberk Özdemir and Sabine Bergler. 2015a. A Comparative Study of Different Sentiment Lexica for Sentiment Analysis of Tweets. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*. INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria, pages 488–496. http://www.aclweb.org/anthology/R15-1064.

Canberk Özdemir and Sabine Bergler. 2015b. CLaC-SentiPipe: SemEval2015 Subtasks 10 B,E, and Task 11. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 479–485. http://www.aclweb.org/anthology/S15-2081.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 1601–1612. http://aclweb.org/anthology/C16-1151.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm Detection on Czech and English Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 213–223. http://www.aclweb.org/anthology/C14-1022.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 451–463. http://www.aclweb.org/anthology/S15-2078.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, D. Christopher Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1631–1642. http://aclweb.org/anthology/D13-1170.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Paris, France.

Emilio Sulis, Delia Iraz Hernndez Faras, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. 2016. Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not . *Knowledge-Based Systems* 108:132 – 143. New Avenues in Knowledge Bases for Natural Language Processing. https://doi.org/10.1016/j.knosys.2016.05.035.

Yong Wang and Ian H. Witten. 1997. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 2449–2460. http://aclweb.org/anthology/C16-1231.