Multilingual Multi-class Sentiment Classification Using Convolutional Neural Networks

Mohammed Attia*, **Younes Samih**†, **Ali Elkahky***, **Laura Kallmeyer**†
*Google Inc.; †University of Düsseldorf
New York, USA; Düsseldorf, Germany
{attia,alielkahky}@google.com
{samih,kallmeyer}@phil.hhu.de

Abstract

This paper describes a language-independent model for multi-class sentiment analysis using a simple neural network architecture of five layers (Embedding, Conv1D, GlobalMaxPooling and two Fully-Connected). The advantage of the proposed model is that it does not rely on language-specific features such as ontologies, dictionaries, or morphological or syntactic pre-processing. Equally important, our system does not use pre-trained word2vec embeddings which can be costly to obtain and train for some languages. In this research, we also demonstrate that oversampling can be an effective approach for correcting class imbalance in the data. We evaluate our methods on three publicly available datasets for English, German and Arabic, and the results show that our system's performance is comparable to, or even better than, the state of the art for these datasets. We make our source-code publicly available.

Keywords: sentiment analysis, sentiment detection, multi-class

1. Introduction

The social media has revolutionized the web by transforming users from being passive recipients of information into contributers and influencers. This has a direct impact on businesses, products and governance. Many of the users' posts are opinions about products and brands that impact other consumers' buying decisions and affect brand trustworthiness. Negative reviews circulated online may cause critical problems for the reputation, competitive power, and survival chances of any business. This in turn has led to some fundamental changes in how businesses approach their customers, gauge satisfaction, provide support and manage risks.

Sentiment analysis is formally defined as the task to identify and analyze subjective information of people's opinions in social media sources (Pham and Le, 2016; Yang et al., 2017). This field of study has recently attracted a lot of attention due to its implications for businesses and governments. The challenges of this task can be summarized in the following points:

- There is a large variety of expressions to denote a range of sentiments. Therefore a sentiment dictionary, no matter how large it could be, cannot list all the possible ways people can express their attitudes.
- Words change meaning depending on the context and the domain. For example, a "short cord" mostly indicates a negative opinion, while a "short boot time" signifies a positive one.
- There could be long-distance dependencies between different constituents of a sentence, and without knowing, for example, the scope of negation, the polarity of an adjective cannot be determined.
- People do not always reveal their opinion in an explicit way, as they could be indirect, subtle or ironical.

Users generally express a broad variety of sentiments with a wide range of degrees, but to simplify the task, sentiment analysis approaches have traditionally classified sentiments into either positive, negative or neutral.

This paper describes our system for multilingual, multi-class sentiment classification using Convolutional Neural Networks (CNNs). We evaluate our system on three datasets in three different languages, and we find that state-of-art results can be achieved without language-specific features or pre-trained word embeddings. We also find that data imbalance has a detrimental effect on multi-class classification. We use over-sampling to balance datasets by repeating instances to increase the size of minority classes, which generally led to significant improvements.

Our methods are language-independent in the sense that we do no rely on ontologies, lists of polarity lexical terms, morphological or syntactic information, thus avoiding the need to deal with out-of-vocabulary opinion words and language-specific features. Our source-code publicly available at `https://github.com/SamihYounes/senti-cnn`

2. Related Work

The different approaches to sentiment analysis share the common theme of mapping a piece of text to a given label from a predefined set (Pang and Lee, 2008). This in essence is similar to some other NLP tasks, such as text categorization (a.k.a. document classification) and language identification. There exists, however, some research on the use of unsupervised methods (Lin and He, 2009).

Research on sentiment analysis is broadly categorized into two paradigms (Cambria, 2016), knowledge-based and statistics-based, depending on whether external language-dependent information and structured resources (such as POS taggers ans polarity lexicons) are included as features in the model or not.

In the first paradigm, machine learning methods are applied to sentiment classification assisted with knowledge-

based features (Mullen and Collier, 2004; Boiy and Moens, 2009; Godbole et al., 2007; Gamon, 2004). Within this domain, work could include a massive amount of prior linguistic knowledge and feature engineering. For example, Wilson et al. (2005) compiled a prior-polarity lexicon of 8,000 subjectivity clues (with around 33% positive, 60% negative and 7% neutral clues) and used 27 linguistic features, including the existence of prior-polarity clues, POS tags, context, use of intensifiers and pronouns, and document topic. This method, however, is unscalable, resource-intensive, and will be hard to adapt to different domains or less-resourced languages.

Some dedicated resources were also created for aiding with this task including the SentiWordNet (Esuli and Sebastiani, 2006; Baccianella et al., 2010) which annotates WordNet synsets according to their degrees of positivity, negativity, and neutrality, and SenticNet (Cambria et al., 2016), which provides the semantic, cognitive and affective information for over 14,000 concepts. Similar lexicons of varying scales were built for some other languages, including, for instance, German (Remus et al., 2010), Arabic (Badaro et al., 2014), and Spanish (Perez-Rosas et al., 2012).

In the second paradigm, using pure statistical methods for sentiment analysis was also successfully applied (Neethu and Rajasree, 2013; Maas et al., 2011; Tripathy et al., 2016). A pioneering work within this approach is the research of Pang et al. (2002) who employed three machine learning methods (Naive Bayes, maximum entropy classification, and support vector machines) to detect polarity in movie reviews. They relied on corpus-driven features using the bag-of-features framework which assumes $f_1, ..., f_m$ as a predefined set of m features (word unigrams and bigrams), $n_i(d)$ as the number of times f_i occurred in a document d, and then each document d is represented by the vector: $\vec{d} := (n_1(d), ..., n_m(d))$. This method is a precursor to the relatively recent word2vec word representation (Mikolov et al., 2013b; Mikolov et al., 2013a; Pennington et al., 2014).

Deep learning for sentiment analysis has also been presented in a number of papers such as (Glorot et al., 2011; Poria et al., 2016; dos Santos and Gatti, 2014). The basic idea with deep learning is to use hidden layers of neural nets to automatically capture the underlying factors that lead from the input to the output, eliminating the need for feature engineering.

3. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) (LeCun et al., 1995) are a powerful deep learning technique because they preserve the spatial structure of the data. They have been shown to produce state-of-the-art results in image processing, computer vision (Krizhevsky et al., 2012) and speech recognition (Graves et al., 2013). In recent years, CNNs have been successfully applied to NLP and document classification problems (Kim, 2014; Johnson and Zhang, 2014).

The input to CNNs is a feature map which corresponds to the pixels in an image or words in a sentence or document, or characters in words. This feature map is scanned in CNNs one area at a time by filters, assuming that filters slide, or convolve, around the feature map. The way CNNs adjust their filter weights is through backpropagation, which means that after the forward pass, the network is able to look at the loss function and make a backward pass to update the weights.

The CNN layer is followed by a pooling layer that compresses or generalizes over the CNN representations. It reduces the dimensionality of the CNN layer by downsampling the output and taking the maximum value as the feature corresponding to each filter.

The pooling layer is typically followed by a feed-forward fully connected layer that takes the features from the pooling layer and makes new combinations for further learning or final predictions.

4. Data Description

4.1. English - The Sanders Twitter Sentiment Corpus

The Sanders Twitter Sentiment dataset[1] (Sanders, 2011) consists of 5,513 tweets related to the products of four companies: Apple, Google, Microsoft, and Twitter. Tweets have been manually tagged as either positive, negative, neutral, or irrelevant with respect to the topic. The distribution of the tagset is in Table 1. We could not find information on how this dataset was annotated, e.g. annotation guidelines, number of annotators involved and whether it was annotated in-house or through crowd-sourcing.

positive	negative	neutral	irrelevant	Total
570	654	2,503	1,786	5,513

Table 1: The Sanders Twitter Sentiment dataset

4.2. German - Deutsche Bahn

The data of the GermEval shared task[2] (Wojatzki et al., 2017) consists of 21,824 messages from various social media and web sources intended for analyzing customer reviews about "Deutsche Bahn", the German public train operator with about two billion passengers annually. The data is split roughly into 90% for training and 10% for development as shown in Table 2. They also provide two test sets. The shared task's Subtask-B is on multi-class document-level polarity, which is about identifying whether the customer's opinion of "Deutsche Bahn" or travel is positive, negative or neutral. The sentence length in the training set ranges between 182 words and just two words. Test set 1 contains larger articles reaching up to 4,666 words in length.

Dataset	positive	negative	neutral
training	1,179	5,048	13,222
development	149	589	1,637
test 1	105	780	1,681
test 2	108	497	1,237

Table 2: Deutsche Bahn Dataset

The data was annotated by a team of six annotators (Wojatzki et al., 2017), and each document was annotated by

[1] http://www.sananalytics.com/lab

[2] https://sites.google.com/view/germeval2017-absa/

positive	negative	mixed	objective	Total
799	1,684	832	6,691	10,006

Table 3: ASTD Dataset

two annotators using WebAnno's curation interface. The documents were checked for consistency by a supervisor who decided on divergences and new issues.

4.3. Arabic - ASTD

The Arabic Sentiment Tweets Dataset (ASTD)[3] (Nabil et al., 2013) consists of 10,006 tweets which are classified as 'positive', 'negative', 'mixed', and 'objective'. The distribution of the tagset is shown in Table 3. The tweets were collected from EgyptTrends and were not related to any particular topic, but generally included comments on diverse political issues. The tweets were annotated through the Amazon Mechanical Turk by three annotators. Tweets that were assigned the same rating by at least two annotators were accepted, otherwise rejected.

A key observation in the three datasets is the noticeable imbalance between the labels as shown in Figure 1, where the "positive" label shows the most disproportionate distribution.

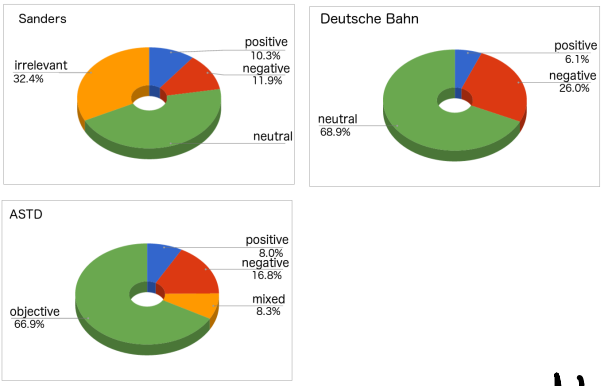

Figure 1: Tagset Distribution

5. System Description

We use a deep neural network model for predicting sentiment polarity. The architecture of our model, shown in Figure 2, is straight-forward. The first layer in our model is a randomly-initialized word embedding layer that turns words in sentences into a feature map and preserves the spatial (contextual) information for each word. This is followed by a convolution neural network (CNN) layer that scans the feature map. This CNN layer has 300 filters and a width of 7, which means that each filter is trained to detect a certain pattern in a 7-gram window of words. Global max-pooling is applied to the output of each filter to take the maximum score of each pattern we search for though the text. The main function of the pooling layer is reduce the dimensionality the CNN representations by down-sampling the output and taking the maximum value as the feature corresponding to each filter. Those score are then supplied to a

[3]http://www.mohamedaly.info/datasets/astd

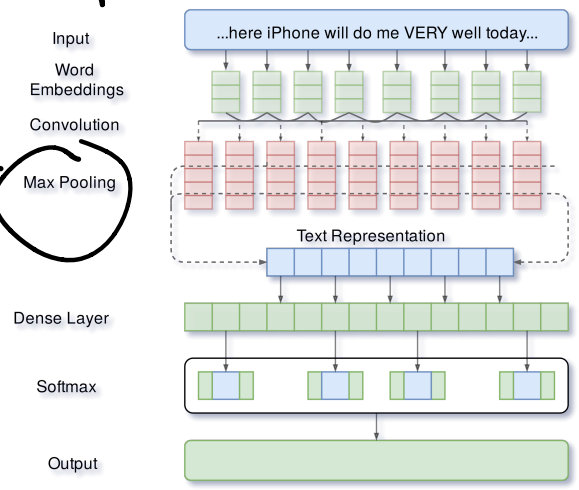

Figure 2: The Architecture of our sentiment detection model applied to an example tweet, best viewed in color. Here the model takes a tweet as its current input to predict its sentiment polarity.

single feed-forward (fully-connected) layer of size 600 and Relu activation to make further learning. Finally, the output of that layer goes through a Softmax layer that predicts the output classes. Table 4 and Figure 2 show the layer configuration and architecture of our model.

Layer	Output Shape	Params
embedding	(None, 300, 300)	10960800
conv1d	(None, 294, 300)	630300
glob_max_pool_1d	(None, 300)	0
dense_1	(None, 600)	180600
dense_2	(None, 3)	2404

Table 4: Neural Network Architecture

For processing the data, we perform manual tokenization on the data by inserting a space between words and punctuation marks. Then we ignore the long tail of the low-frequency words (heuristically setting the threshold at 3, i.e. ignoring words that occur three times or less), and remove URL addresses.We tried ignoring the top three most frequent words, assuming that they are semantically-irrelevant function words (e.g. punctuations, determiners and prepositions), but this led to lowering the performance.

6. Experiments and Results

We compare our system to the best scores published in the literature for each dataset. We consider the comparison meaningful only when there is a test set provided. Otherwise, we create our own splits of the data, and the comparison with previously published results are only indicative.

6.1. Results on Sanders Dataset

Sentiment analysis for English using the Sanders dataset has been reported in a number of papers. For example, Bravo-Marquez et al. (2013) extracted features from a number of lexical resources (such as OpinionFinder, AFINN and SentiWordNet) and applied a number of traditional ML classifiers (such as J48, Naive Bayes, Logistic

and SVM), and reported an accuracy of 70.1%. Da Silva et al. (2014) used the Ensemble method (training multiple learners to solve the same problem) and reported an accuracy score of 76.25%. Similarly Hassan et al. (2013) used a stack of learning models and a voting mechanism and obtained an accuracy of 76.30%. Using our system we achieved a significantly higher accuracy score of 78.3%. However, as there were no dedicated test set for this dataset we tested our system on a randomly selected subset of 20% of the data, which makes the comparison with the other systems not conclusive.

It is to be noted that Johansson and Lilja (2016) reported a higher score for the Sanders dataset of 84.38% using a lexicon, but they reported that the method did not scale well to other datasets including IMDB and Sentiment140. It is also not clear to us whether their experiments were limited to the binary polarity of positive and negative or they predicted the full range of classes.

6.2. Results on Deutsche Bahn

The Deutsche Bahn dataset has been used in the GermEval-2017 Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. The best results on Test 1 were reported by Naderalvojoud et al. (2017), Hövelmann and Friedrich (2017) and Sidarenka (2017) who achieved accuracy scores of 74.94%, 74.786% and 74.47% respectively. Hövelmann and Friedrich (2017) used three German lexicons (combined in what they called the German SentiWordNet Lexicon) and fed them to a deep recurrent neural network (RNN) classifier to discover context-based sentiments. Hövelmann and Friedrich (2017) on their turn used a fastText classifier (the Facebook open-source library for text representation), enhanced with pretrained vectors and gradient boosted trees (GBTs) trained on bag-of-words (BOWs). Sidarenka (2017) used a hybrid approach by joining an SVM module trained on user-specified attributes with a bi-directional LSTM network.

We fine tune the hyper parameters of our system on the development set of the GermEval shared task. Our system gives an accuracy of 75.45% on Test 1 which is slightly above the results of the best system. We consider the comparison here meaningful as we are testing against the same benchmark set.

6.3. Results on the ASTD Dataset

The ASTD dataset has been used in (Nabil et al., 2013) and the best reported accuracy score is 69.1% using an SVM classifier. They tried to balance the data using undersampling which did not perform as well. Our system gives an accuracy of 67.93%, which is slightly below their results, but the evaluation is not conducted on the same set. As no test set was provided we randomly selected a 20% subset from the available data.

7. Dealing with Data Imbalance

We notice that the prediction results favor the majority classes at the expense of the minority classes as the system attempts to achieve high accuracy scores. Trying to resolve this issue, we need first to find an evaluation metrics that gives equal weight to the classes in the labelset, and second find a way to balance the data. For the evaluation measure, we choose the F1 score with a macro average as it calculates the f-score for each label, and outputs their unweighted mean, allowing each class to have the same weight as the other classes regardless of the number of instances.

To balance out the training data, we apply manual oversampling by repeating records of the minority classes. It is to be noted that SMOTE (Synthetic Minority Over-Sampling Technique) is sometime successfully applied to numerical data, but it is not applicable to textual data as is case with the data here. We compare the results using the macro F1 before and after oversampling for the three datasets.

For the English dataset, Tables 5 and 6 show the confusion matrix for the model's predictions without and with oversampling respectively. In oversampling we repeated the records for 'Positive' and 'Negative' three-folds. The oversampling technique works in improving the performance raising both the accuracy (from 78.60% to 79.57%) and the macro F-measure scores (from 69.13% to 70.23%).

	Negative	**Positive**	**Neutral**	**irrelevant**
Negative	56	12	39	0
Positive	10	50	36	5
Neutral	29	37	384	23
irrelevant	3	1	24	314
Accuracy	78.60			
F1 Macro	69.13			

Table 5: Confusion Matrix for the Sanders Dataset without Oversampling

	Negative	**Positive**	**Neutral**	**irrelevant**
Negative	55	4	46	2
Positive	11	47	39	4
Neutral	25	21	404	23
irrelevant	3	1	30	308
Accuracy	79.57			
F1 Macro	70.23			

Table 6: Confusion Matrix for the Sanders Dataset with Oversampling

For the German dataset, Table 7 show the confusion matrix for the model predictions for the test set without oversampling. Table 8 shows the results after oversampling 'Positive' and 'Negative' three folds. The oversampled model shows a better confusion matrix with an increase in the macro F1 score (from 49.00% to 45.84%) despite the decrease in Accuracy.

	Negative	**Positive**	**Neutral**
Negative	344	1	435
Positive	10	4	91
Neutral	89	4	1588
Accuracy	75.45		
F1 Macro	49.00		

Table 7: Confusion Matrix for the Deutsche Bahn Dataset without Oversampling

	Negative	Positive	Neutral
Negative	482	5	293
Positive	16	16	73
Neutral	259	28	1394
Accuracy	73.73		
F1 Macro	54.84		

Table 8: Confusion Matrix for the Deutsche Bahn Dataset with Oversampling

For the Arabic dataset, Table 9 show the confusion matrix without oversampling, and Table 10 shows the results after oversampling by repeating 'Positive' and 'Mixed' three folds, and double sizing the 'Negative'. The oversampled model again shows a better macro F1 score with an increase from 29.82% to 35.32%.

	Negative	Positive	Objective	Mixed
Negative	111	0	233	0
Positive	6	0	167	0
Objective	73	0	1249	0
Mixed	43	0	120	0
Accuracy	67.93			
F1 Macro	29.82			

Table 9: Confusion Matrix for the ASTD Dataset without Oversampling

	Negative	Positive	Objective	Mixed
Negative	89	2	237	16
Positive	10	16	145	2
Objective	30	73	1191	28
Mixed	24	4	118	17
Accuracy	65.58			
F1 Macro	35.32			

Table 10: Confusion Matrix for the ASTD Dataset with Oversampling

Oversampling the data, simply by replicating records, only tricks the model into giving greater weight to a certain class, but does not provide any essentially useful information to the system to base it's judgment on. Therefore, we recommend data annotation to selectively target more data to increase the size of the minority classes to allows the system to better understand and predict these classes.

8. Conclusion

In this paper we have presented our systems for multi-class sentiment classification and we found that the deep neural network model can outperform traditional methods that rely on language-specific feature engineering. We show that the class imbalance in the data can lead to degradation in the system performance, and point out that oversampling can be a helpful workaround for handling this imbalance.

Further we note that the system performs better on the Sanders dataset, followed by the Deutsche Bahn and the ASTD datasets. This could be connected to the observation that the ASTD dataset has the largest number of classes (5 classes compared to 4 in Sanders and 3 in Deutsche Bahn) and there could be an inverse correlation between the increased number of classes and the system performance.

Beside the class balance and the granularity of classes, we assume that the system performance could be also impacted by the quality of the annotation (in-house vs. crowdsourced) and whether the polarity is related to a specific or general topic.

9. Bibliographical References

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Badaro, G., Baly, R., Hajj, H., Habash, N., and El-Hajj, W. (2014). A large scale arabic sentiment lexicon for arabic opinion mining. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 165–173.

Boiy, E. and Moens, M.-F. (2009). A machine learning approach to sentiment analysis in multilingual web texts. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 12:526–558.

Bravo-Marquez, F., Mendoza, M., and Poblete, B. (2013). Combining strengths, emotions and polarities for boosting twitter sentiment analysis. In *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*, Chicago, USA.

Cambria, E., Poria, S., Bajpai, R., and Schuller, B. (2016). Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. *In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2666–2677.

Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31:102–107.

Da Silva, N. F., Hruschka, E. R., and Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, 66:170–179.

dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. *In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 417–422.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. *In Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. *In Proceedings of the Twenty-eight International Conference on Machine Learning, ICML*.

Godbole, N., Srinivasaiah, M., and Skiena, S. (2007). Large-scale sentiment analysis for news and blogs. *In*

ICWSM 2007 - International Conference on Weblogs and Social Media.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Hassan, A., Abbasi, A., and Zeng, D. (2013). Twitter sentiment analysis: A bootstrap ensemble framework. In *Social Computing (SocialCom), 2013 International Conference on*, pages 357–364. IEEE.

Hövelmann, L. and Friedrich, C. M. (2017). Fasttext and gradient boosted trees at germeval-2017 tasks on relevance classification and document-level polarity. In *Proceedings of the GermEval 2017 Shared Task*, Berlin, Germany.

Johansson, H. and Lilja, A. (2016). Method performance difference of sentiment analysis on social media databases: Sentiment classification in social media.

Johnson, R. and Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. *In Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 375–384.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., , and Potts, C. (2011). Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 142–15.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *In Proceedings of International Conference on Learning Representations (ICLR) 2013. arXiv:1301.3781v3*, pages 746–751, Scottsdale, AZ.

Mikolov, T., tau Yih, W., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT 2013*, pages 746–751, Atlanta, Georgia.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. *Proceedings of EMNLP-04, 9th Conference on Empirical Methods in Natural Language Processing*.

Nabil, M., Aly, M., and Atiya, A. (2013). Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 2515–2519, Lisbon, Portugal.

Naderalvojoud, B., Qasemizadeh, B., and Kallmeyer, L. (2017). Hu-hhu at germeval-2017 sub-task b: Lexicon-based deep learning for contextual sentiment analysis. In *Proceedings of the GermEval 2017 Shared Task*, Berlin, Germany.

Neethu, M. S. and Rajasree, R. (2013). Sentiment analysis in twitter using machine learning techniques. pages 1–5, July.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. *EMNLP 2002*, 10:79–86.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Perez-Rosas, V., Banea, C., and Mihalcea, R. (2012). Learning sentiment lexicons in spanish. In *LREC*, volume 12, page 73.

Pham, D.-H. and Le, A.-C. (2016). A neural network based model for determining overall aspect weights in opinion mining and sentiment analysis. *Indian Journal of Science and Technology*, 9:1–6.

Poria, S., Chaturvedi, I., Cambria, E., and Hussain, A. (2016). Convolutional mkl based multimodal emotion recognition and sentiment analysis. *In Proceedings of the Twenty-eight International Conference on Machine Learning, ICML*, pages 439–448.

Remus, R., Quasthoff, U., and Heyer, G. (2010). Sentiws-a publicly available german-language resource for sentiment analysis. In *LREC*.

Sanders, N. J. (2011). Sanders-twitter sentiment corpus. *Sanders Analytics LLC*.

Sidarenka, U. (2017). Potts at germeval-2017 task b: Document-level polarity detection using hand-crafted svm and deep bidirectional lstm network. In *Proceedings of the GermEval 2017 Shared Task*, Berlin, Germany.

Tripathy, A., Agrawal, A., and Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *In Expert Systems With Applications*, 57:117–126.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. pages 347–354.

Wojatzki, M., Ruppert, E., Holschneider, S., Zesch, T., and Biemann, C. (2017). Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback. In *Proceedings of the GermEval 2017 Shared Task*, Berlin, Germany.

Yang, K., Cai, Y., Huang, D., Li, J., Zhou, Z., and Lei, X. (2017). An effective hybrid model for opinion mining and sentiment analysis. *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 465–466.