UKP TU-DA at GermEval 2017: Deep Learning for Aspect Based Sentiment Detection

Ji-Ung Lee, Steffen Eger, Johannes Daxenberger, Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
Ubiquitous Knowledge Processing Lab (UKP-DIPF)
German Institute for Educational Research and Educational Information
http://www.ukp.tu-darmstadt.de

Abstract

This paper describes our submissions to the GermEval 2017 Shared Task, which focused on the analysis of customer feedback about the Deutsche Bahn AG. We used sentence embeddings and an ensemble of classifiers for two sub-tasks as well as state-of-the-art sequence taggers for two other sub-tasks. Relevant aspects to reproduce our experiments are available from `https://github.com/UKPLab/germeval2017-sentiment-detection`.

1 Introduction

For many companies, customer feedback is an important source for identifying problems affecting their services. Although customer feedback may be obtained by interviewing single customers or conducting larger studies using questionnaires, those are often cost-intensive. Instead, it is much cheaper to crawl customer feedback from the web, for example from social media platforms like Twitter, Facebook, or even news pages. In contrast to interviews or questionnaires, crawled data is often noisy and does not necessarily cover specific company-related topics. Due to the vast amount of available data on the web, it is crucial to analyze relevant documents and extract the feedback automatically.

The GermEval 2017 Shared Task (Wojatzki et al., 2017) focuses on the automated analysis of customer feedback about the *Deutsche Bahn AG* (DB) in four subtasks, namely (A) relevance classification of documents, (B) identification of the document-level polarity, (C) identification of certain aspects in a single document as well as predicting their category and polarity, and (D) extraction of the exact phrase of a single aspect.

For example, the tweet

> *@RMVdialog hey, wann fährt denn nach der Störung jetzt die nächste Bahn von Glauberg nach Ffm?*

has the following gold-standard annotations for Tasks A-D, respectively:

(A)	true
(B)	neutral
(C)	Sonstige Unregelmässigkeiten – negative
(D)	Störung

We participated in all subtasks of the shared task. For Tasks A, B, and C we trained models on document-level representations using a classifier ensemble. As Task D can be modeled as a sequence tagging task, we used a state-of-the-art deep neural network tagger with a conditional random field at the output layer.

This work is structured as follows. Section 2 gives an overview of the data. Section 3 details the two main modeling approaches we made use of. Section 4 describes our experimental set-ups, and presents and discusses our results for a selection of well-performing models. We conclude in Section 5.

2 Data

The data provided for this shared task contains ≈ 22,000 German messages from various social media and web sources and has been annotated in a joint project between Technische Universität Darmstadt and DB. In addition to the provided data we used several external resources for training word and sentence embeddings and computing task specific features.

2.1 Task Specific Data

The shared task data contains annotations about the relevance R of a message (Task A) and its sentiment polarity (B), either positive P, negative NG, or neutral NT. Relevant messages contain further annotations about their aspects, with the aspect category and sentiment polarity (C) and its exact

	R (A)	P (B)	NG (B)	NT (B)
Total	83	6	26	68

Table 1: Class distributions for task A and B in %

Category	#
Allgemein	13892
Zugfahrt	2421
Sonstige Unregelmässigkeiten	2112
Atmosphäre	1576
Sicherheit	962
Ticketkauf	741
Service und Kundenbetreuung	551
Connectivity	390
Informationen	388
Auslastung und Platzangebot	304
DB App und Website	252
Komfort und Ausstattung	166
Barrierefreiheit	89
Toiletten	54
Image	54
Gastronomisches Angebot	47
Reisen mit Kindern	46
Design	37
Gepäck	15
QR-Code	1
Total	24098

Table 2: The number of aspects per category.

phrase (target) identified by the character offsets (D). Table 1 shows the distribution of classes in the train and dev sets for tasks A and B.

For Tasks C and D, the data contains 24,098 aspects in total, which are classified into 20 different categories. Table 2 shows the number of aspects for each category. We observe that the data is highly skewed here, with more than 57% of all aspects being of category “Allgemein”. Table 3 shows the distribution of positive, negative, and neutral aspects in the data.

Furthermore, not every aspect can be matched to an exact phrase (target). In 44% of the cases, a category and the polarity is assigned to a message without having a target. For these cases, the target is annotated by NULL.

	P (C)	NG (C)	NT (C)
Total	10	42	48

Table 3: Polarity distribution of aspects in %

2.2 External Sources

We use several external data sources for training various word and sentence embeddings, namely a German Wikipedia corpus (Al-Rfou et al., 2013) and a German Twitter corpus (Cieliebak et al., 2017). The Wikipedia corpus is publicly available and contains already tokenized data. We use a crawler published along with the Twitter corpus, to obtain the actual texts of the tweets. This results in a corpus containing 7464 tweets, which we then tokenized using the Tweet Tokenizer from NLTK (Bird et al., 2009).

We also made use of an English Twitter sentiment corpus of around 40K tweets (Rosenthal et al., 2017), each annotated with positive, negative, or neutral stance, just as the German data. Our hope was that this would provide a strong additional signal from which our learners could induce the sentiment of a tweet, be it English or German. To make use of this additional data, we projected our word and sentence embeddings (see below) in a bilingual German-English embedding space so that they are comparable.[1] We used CCA (Faruqui and Dyer, 2014) for this, which requires independently constructed language specific embeddings and word translation pairs (such as (*Katze*,*cat*)) to allow projecting vectors into a joint space. The word translation pairs were induced from the Europarl corpus (Koehn, 2005).

3 Methods

In what follows, we describe, on a general level, our approaches to Tasks A and B (Section 3.1) and Task D (Section 3.2), respectively. For Task C, we mixed between the approaches outlined in Sections 3.1 and 3.2 in our experiments. We relegate the corresponding model description to Section 4.

3.1 Sentence Embeddings and Classifier Ensemble

We used a unified and minimally expensive (in terms of feature engineering) approach to tackle Tasks A and B, which both concern the classification of documents into categories. We tokenized

[1] Besides using the English Twitter sentiment corpus for computing word embeddings, we had hoped that the annotated English data would improve our classification results in German, but initial experiments in which we (naively) merged both annotated datasets led to performance deteriorations, so we abandoned the idea.

each document and converted it to an embedding via the tools Sent2Vec (Pagliardini et al., 2017) and SIF (Arora et al., 2017). Both of these tools aspire to improve upon the simple average word embedding baseline for sentence embeddings, but are conceptually simple. We trained Sent2Vec on the union of German Wikipedia data as well as a Twitter corpus and the task specific data of the Shared Task. For SIF, we first created word embeddings with the standard skip-gram model of Word2Vec (Mikolov et al., 2013), and then generated sentence embeddings from these via specific SIF parametrizations outlined below. We train Word2Vec on the same data sources as Sent2Vec.

After converting documents to embeddings of particular sizes d, we train a classifier that maps representations in $\mathbb{R}^d$ to one of N classes, where $N = 2$ for Task A and $N = 3$ for Task B. We use the stacked learner from Eger et al. (2017) as a classifier. This is an ensemble based system that uses several base classifiers from scikit-learn and a multilayer perceptron as a meta-classifier to combine the predictions of the base classifiers.

3.2 (MTL) Sequence Tagging

Task D is naturally modeled as sequence tagging task, that is, it can be framed as the problem of tagging each element in a sequence of tokens $x_1, \ldots, x_T$ with a label $y_1, \ldots, y_T$. We used the most recent state-of-the-art sequence tagging frameworks (Lample et al., 2016; Ma and Hovy, 2016), which consist of a neural network (bidirectional) LSTM tagger that uses word and character level information as well as a CRF layer on top that accounts for dependencies between successive output predictions. Moreover, since multi-task learning (MTL) settings in which several tasks are learned jointly have been reported to sometimes outperform single-task learning (STL) scenarios, we directly allow for inclusion of several tasks during training and prediction time. Our approach builds here upon the architecture of Søgaard and Goldberg (2016) in which different tasks feed from particular levels of hidden layers in a deep LSTM tagger. Our employed framework (Kahse, 2017) extends Søgaard and Goldberg (2016) in that we include both character and word-level information as well as implement CRF layers for each task, as mentioned already. Note that we could in principle train all four Shared Task tasks in a single architecture, possibly with Tasks A and B feeding from lower layers of the deep LSTM, because the tasks satisfy some of the requirements that have often been attributed to successful MTL, such as relatedness of tasks and natural task hierarchy.

To illustrate, for Task D, the goal is to extract the relevant phrase to be classified in Task C. We frame this as a token-level BIO tagging problem in which each token is labeled with one of three classes from $\{I, O, B\}$. That is,

Notrufsystem	:	250	Funklöcher	bei	...
B	I	I	I	O	...

retrieves the target phrase *Notrufsystem : 250 Funklöcher* from the document.

4 Experiments

Baseline: The organizers of the shared task provided baselines, consisting of an SVM with unigram word features for Tasks A, B, and C and a CRF for Task D.[2]

4.1 Tasks A and B

Approach: We train models with document-level features using the stacked learner. We focus on the comparison of different word and sentence embeddings, and additional polarity features computed using a lexical resource described in Waltinger (2010) for Task B. For document embeddings, we evaluate average word vectors, besides the approaches mentioned above. Furthermore, we ran experiments with combinations of different word and sentence embeddings. For these, we compute average word vectors for a single document and concatenate it with the respective sentence embedding.

Hyperparameters: We compare Word2Vec and 100 dimensional Komninos word embeddings (Komninos and Manandhar, 2016), and 500 dimensional Sent2Vec and SIF sentence embeddings as described before.[3] Word2vec skip-gram embeddings are computed for dimensions $d = 50, 100, 500$. In addition, we compare two SIF embeddings computed with different input word embeddings. One was computed from the German data directly and another one by projecting the

[2] An updated data set was released on August, 10th. For our submissions, we retrained all models on the new data.

[3] For computing the SIF embeddings, we use the word weighting parameter $a = 0.01$ and, we subtract the first $r = 2$ principal components. See the original paper for details.

	Micro F1
Baseline	0.882
W2V ($d = 50$)	0.883
W2V ($d = 500$)	**0.897**
S2V	0.885
S2V + W2V ($d = 50$)	0.891
S2V + K + W2V($d = 50$)	0.890
SIF (DE)	0.895
SIF (DE-EN)	0.892

Table 4: Task A results

	Micro F1
Baseline	0.709
W2V ($d = 50$)	0.736
W2V ($d = 500$)	0.753
S2V	0.748
S2V + W2V ($d = 50$)	0.744
S2V + K + W2V($d = 50$)	0.749
SIF (DE)	0.759
SIF (DE-EN)	**0.765**

Table 5: Task B results

	Micro F1
Baseline	0.709
W2V ($d = 50$)	0.748
W2V ($d = 500$)	0.756
S2V	0.748
S2V + W2V ($d = 50$)	0.755
S2V + K + W2V($d = 50$)	0.751
SIF (DE)	0.748
SIF (DE-EN)	**0.757**

Table 6: Task B results with polarity features

	Macro F1
Baseline	0.478
MTL_{Adam} ($d = 50$)	0.438
STL_{Adam} ($d = 50$)	0.458
STL_{Adam} ($d = 100$)	0.488
STL_{Adam} ($d = 100$) + POS-Tags	0.494
$\text{STL}_{\text{AdaDelta}}$ ($d = 100$)	0.543
$\text{STL}_{\text{AdaDelta}}$ ($d = 100$) + POS-Tags	**0.554**

Table 7: Task D results

German data into a shared embedding space with English embeddings as described before.

Results: The results of the models better than the baseline are reported in Tables 4 and 5. As can be seen, all models only slightly outperform the baseline in Task A. For Task B, all models trained on the stacked learner beat the baseline substantially even when using only plain averaged word embeddings. We furthermore trained models on additional polarity features for Task B as mentioned before. For this, we look up all positive, negative, and neutral words in a document and compute a three-dimensional polarity vector by using the total count of found words. These are concatenated to the respective document representation. Adding the polarity features improved the results for all models except for those using SIF embeddings (Table 6).

Discussion: Unexpectedly, the model using the averaged Word2Vec embedding performs best for Task A, even though the other embeddings created by Sent2Vec or SIF have the same dimension (500). A reason for this may be chance or the Twitter data. As the experiments of Pagliardini et al. (2017) confirm, averaged Word2Vec embeddings perform rather well for a similarity task on Twitter data. However, we observe that particularly SIF outperforms average word embeddings for Task B. We also observe that the joint EN-DE embeddings improve results for Task B (+0.6% and +0.9%, respectively) but lead to a drop in performance for Task A (-0.3%). This is in line with the common observation that the bilingual signal may provide an additional source of both useful and noisy, irrelevant, or even hurtful information (Faruqui and Dyer, 2014; Eger et al., 2016).

4.2 Task D

Approach: We tackle this task with our sequence tagging framework and evaluate on the dev set using the macro F1 score.

Hyperparameters: We use Word2Vec embeddings of $d = 50, 100$ trained on German Wikipedia, Twitter, and the shared task data. We also incorporate 20 dimensional skip-gram embeddings for POS-tags, trained on the data of the shared task and concatenate them with the corresponding word vectors. The German STTS POS-Tags were computed with the Marmot POS-Tagger (Müller et al., 2013). We furthermore compute 30 dimensional character embeddings on the shared task data (i.e., not pre-trained), using an LSTM with 50 hidden units. Dropout is set to 0.2 for the BLSTM and the batch size is set to 50 for all experiments. All models were trained with 100 hidden units.

Results: Since the evaluation tool provided by the task organizers always requires a category for computing the scores on Task D, we evaluated our systems using the macro F1 score on the BIO tags. For comparison with the baseline, we compute the score by converting the predictions into BIO format. Table 7 contains our results for Task D.

We trained different set-ups with STL and MTL models. First of all, we evaluated STL against MTL by training two models on 50 dimensional Word2Vec embeddings. For the MTL set-up, we defined the BIO tagging (D) as the main task and added tasks A, B, and C as auxiliary tasks. For document-level annotations (Task A and B) each token of the document is tagged with the respective class of the document. As the results show, the MTL set-up did not improve the macro F1 score in this setting. Thus, we tried to improve the predictions of the STL model in our follow-up experiments. The best results were achieved by using 100 dimensional Word2Vec embeddings with additional POS-Tag embeddings. Furthermore, using AdaDelta as an optimizer yielded better results than using Adam.

Further results, using the organizers' evaluation tool, can be found below.

4.3 Task C

Approach: There are several difficulties for this task. First, documents may contain several aspects of different categories, making this at least a multi-class classification problem for document-level approaches. Furthermore, in some cases one document contains several aspects of the same category. On a document-level, one either has to give up on predicting multiple aspects of one class, or add classes for each possible combination of categories, leading to a huge number of classes which do not scale well to new data. Second, there exist aspects with NULL targets which are not assigned to any tokens in the text, but still belong to a category and have a polarity. They cannot be expressed properly on a token-level, as they were not annotated with this intention. One solution may be assigning all tokens of a document to a NULL target category, but this leads to overlapping categories on a token-level, adding more difficulty to the task itself.

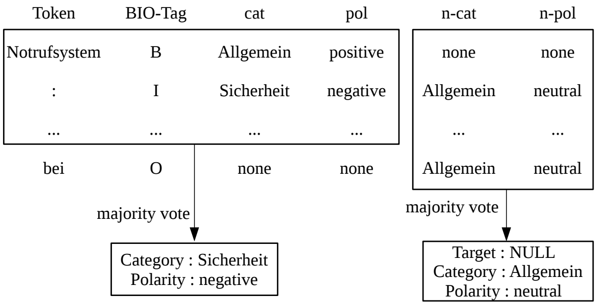

Figure 1: Combination of predictions from several independent sequence tagging models (Task C).

To obtain aspect category and polarity predictions, we evaluate various combinations of the stacked learner and the sequence tagger. We report the results for three approaches. (1) We use independent STL sequence taggers to predict BIO labeling as well as category and polarity of each token in a document (`INDEP`). (2) We predict BIO labeling first, and feed each identified entity to our described ensemble model to predict category and polarity of the identified targets (`PIPE`). (3) We use the Sequence Tagger for BIO tagging and category prediction (label set is $\{B,I,O\} \times \{\text{Allgemein}, \text{Sicherheit}, \ldots\}$) and the stacked learner for polarity prediction (`JOINT`).

`INDEP`: We train a separate model for five subtasks, namely the prediction of BIO labeling, category (cat), polarity (pol), NULL category (n-cat), and NULL polarity (n-pol). If the BIO model predicts B or I for a given token, we look up the cat and pol prediction and obtain the final prediction via a majority vote over the span of BI tokens. Since O tokens are mapped to the none class, we only predict category and polarity if both are present. As the n-cat and n-pol predictions do not depend on the BIO prediction, we perform a majority vote over the whole document. Figure 1 shows an example of how we combine the predictions for the individual subtasks from different STL models.

`PIPE`: We train models with the stacked learner for aspect categories and their polarity. As the BIO predictions do not include NULL targets, we train a separate model on the binary task whether or not a document contains a NULL target. Instead, one could also add another class for documents without any aspects, however we decided not to increase the difficulty for Task C as it already contains 20 classes. If a document is predicted to contain a NULL target, it is added as input for category and polarity prediction. Figure 2 shows the interaction of all models and how the predictions are forwarded

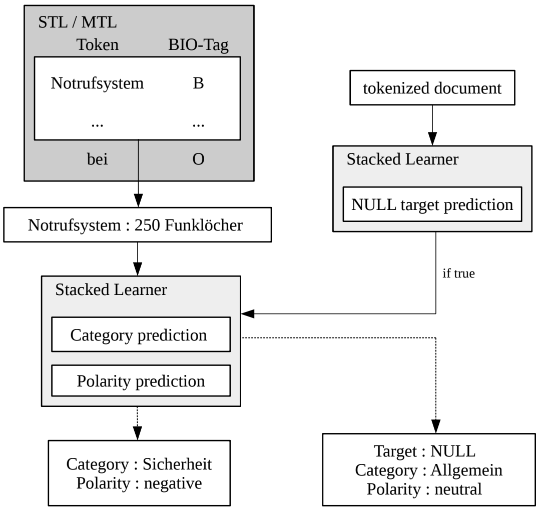

Figure 2: Prediction of category and polarity using a pipeline of stacked learner models (Task C).

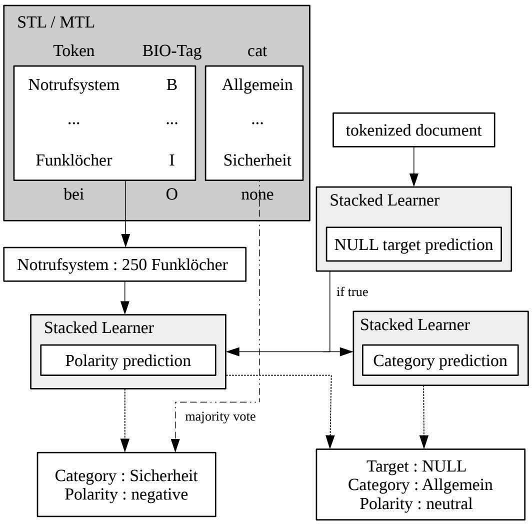

Figure 3: Computing predictions for using STL and SL predictions (Task C).

to the next model for prediction.

JOINT: Here, we use the BIO and category predictions of the sequence tagger, while using the stacked learner for polarity prediction. For NULL targets, we train a separate model for category prediction on the stacked learner similar to the PIPE approach. The model is illustrated in Figure 3.

Results: We train the stacked learner using 500 dimensional Word2Vec embeddings, which showed a good performance for tasks A and B. We do not use Sent2Vec or SIF, since many targets for category and polarity prediction consist of only one word. For targets of longer sequences, we average the word vectors over all tokens.

We used the same hyper-parameters as in Tasks A, B, and D. Table 8 shows the results for our three final systems (INDEP, PIPE, and JOINT) evaluated with the tool provided by the organizers. It calculates the micro F1 scores of only the categories (C-1) and the categories along with their sentiment (C-2) for Task C, and for Task D the micro F1 scores based on exact (D-1) and overlapping (D-2) matching of the offsets. We obtain BIO predictions for Task D using the best model, namely $\text{STL}_{\text{AdaDelta}}$ (d = 100 + POS-Tags), and trained the additional models for the INDEP and JOINT approaches with the same parameters.

As can be seen, INDEP consistently outperforms the baseline except for C-1. Further, PIPE outperforms INDEP except for D-1, where it performs even worse than the baseline. The JOINT approach lies between INDEP and PIPE on average.

Strangely, the organizers' evaluation tool includes the category prediction from Task C for calculating the scores of Task D. The reason for this may be a different point of view for Tasks C and D. If one first identifies the targets and predicts the category and sentiment accordingly, the score for Task D should not be affected by the results for Task C. However, if one first predicts all categories and their sentiment in a document and identifies the targets afterwards, it is important to map the targets to their appropriate categories. Then the correct mapping of category and target may be seen as an additional task which has to be considered for calculating the score for Task D. So even if INDEP, PIPE and JOINT have the same BIO output for Task D, their scores differ due to different predictions of category and sentiment. For example, if the sequence tagging model for categories predicts none for a given chunk, the JOINT and INDEP model discard it for the final results, leading to a different score with the provided evaluation tool. While we tried to model the tasks as they were introduced, our approach to first identify the targets and then to predict category and sentiment seems more intuitive. This way, we do not have the problem of dealing with multiple assignments of one category for a document, as the task is solved on a token-level with a distinct label.

Discussion: All three approaches fail to predict any of the categories *Design*, *Image*, and *QR-Code*. In addition, the JOINT model did not predict any of the categories *Gastronomisches Angebot*, *Toiletten*, *Reisen mit Kindern*, and *Gepäck*. The INDEP model predicted the least number of different cate-

	C-1	C-2	D-1	D-2
Baseline	**0.477**	0.334	0.244	0.329
INDEP	0.429	0.377	**0.253**	0.364
PIPE	0.476	**0.381**	0.233	**0.386**
JOINT	0.443	0.367	0.250	0.377

Table 8: Task C and D results calculated with the provided evaluation tool

gories, adding *Informationen*, *Barrierefreiheit*, and *Auslastung und Platzangebot* to those mentioned before. This is unsurprising given that some categories occur very infrequently in the data (cf. Table 2) and the general skewness of the data distribution.

5 Conclusion

We presented our submissions to the GermEval 2017 Shared Task, which focused on the analysis of customer feedback about the Deutsche Bahn AG. We used neural sentence embeddings and an ensemble of classifiers for two sub-tasks as well as state-of-the-art sequence taggers for two other subtasks. We substantially outperformed the baseline particularly for Task B, the detection of sentiment in customer feedback, as well as for Task D, the extraction of phrases which carry category and polarity of a meaningful aspect in customer feedback.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 03VP02540 (ArgumenText) and 01UG1416B (CEDIFOR).

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed Word Representations for Multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *International Conference on Learning Representations*, April.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition.

Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli. 2017. A twitter corpus and benchmark resources for german sentiment analysis. In *Proceedings of the 4th International Workshop on Natural Language Processing for Social Media (SocialNLP 2017)*, pages 45–51. Association for Computational Linguistics.

Steffen Eger, Armin Hoenen, and Alexander Mehler. 2016. Language classification from bilingual word embedding graphs. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, page (to appear), December.

Steffen Eger, Erik-Lân Do Dinh, Ilia Kutsnezov, Masoud Kiaeeha, and Iryna Gurevych. 2017. EELECTION at SemEval-2017 Task 10: Ensemble of nEural Learners for kEyphrase ClassificaTION. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, Vancouver, Canada, August. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. Improving Vector Space Word Representations Using Multilingual Correlation. In *Proceedings of the European Association for Computer Linguistics*.

Tobias Kahse. 2017. Multi-Task Learning for Argumentation Mining. Master Thesis.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500, San Diego, California, June. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1: Long Papers, pages 1064–1074, Berlin, Germany, August. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October. Association for Computational Linguistics.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *CoRR*, abs/1703.02507.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, Vancouver, Canada, August. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Ulli Waltinger. 2010. German polarity clues: A lexical resource for german sentiment analysis. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, May. electronic proceedings.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback. In *Proceedings of the GSCL GermEval Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, Berlin, Germany.