**1. BERT**

The second part of our work is based on the Bidirectional Encoder Representations from Transformers (BERT) and its variations. The name comes from its core architecture as a multi-layer bidirectional Transformer encoder. (https://arxiv.org/pdf/1810.04805.pdf)

The Transformer-based models especially in the recent NLP research consist of transfer learning and attention. In standard supervised learning approach, where we ideally have a large number of labelled training instances, we assume the test data to be drawn from the same distribution in order to test the goodness of generalization in a reliable way. Yet, given another domain, for example another set of documents with different thematic context, we cannot expect our model to generalize well.

In many situations, as well as in our data situation, one has to (manually) label a large amount of data instances, which is not practicable at all. So, in data situations where the number of annotated training documents is restricted, it becomes hard to deal with the highly resource intensive supervised learning. Therefore, it is appropriate to apply the transfer learning that represents the approach of statistical learning in which we firstly train a model on an original task and domain and then transfer this learned knowledge to the target task and domain. The detailed workflow of Transformer is described in a paper by researchers at Google AI Language.

(Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, <https://ieeexplore.ieee.org/document/5288526> )

(Ruder, S. (2019). Neural transfer learning for natural language

processing. PhD thesis, National University of Ireland.

https://ruder.io/thesis/neural transfer learning for nlp.pdf.)

Moreover, the Transformer based architectures do not process the sequential data (such as texts) in order, by utilizing the mechanism of attention. Due to this property, these architectures allow for much more parallelization than a standard RNN and leads to reduced training times. (<https://arxiv.org/pdf/1706.03762.pdf>)

This is mainly achieved in the pretraining task of a BERT model on a large corpus in a specific language with the help of neural networks. In the fine-tuning step, this knowledge (so the parameter values learned during training the source model) is then applied to a new purpose-specific context to learn a certain task and vocabulary. (Devlin, 2019)

**1.1 Input Preprocessing**

*Die Ausgrenzung von MigrantInnen von der #EssenerTafel ist inakzeptabel und rassistisch. Wir dürfen nicht zulassen, dass die Ärmsten gegeneinander ausgespielt werden.*

In order to apply BERT, one has to process the textual input to the encoder of BERT. For that, we split the words into tokens or wordpieces based on a given vocabulary, which is determined by the pretrained language model. Then convert each sequence of tokens into numerical vectors, embeddings and process them in the neural network with some additional tokens: The beginning of the first sequence is signed by a “[CLS]” token, the end by a “[SEP]” token. Due to a marker Sentence A or B to each token in a sequence helps to indicate which sentence a token belongs in. Moreover, a positional embedding stands for the position of each token in the sentence. The inputs have to be of same length, whereby the maximum length of an input sequence can be 512 tokens. Shorter inputs are filled with padding tokens and the longer ones are truncated.

***[CLS]*** *Die Ausgrenzung von MigrantInnen von der # EssenerTafel ist inakzeptabel und rassistisch. Wir dürfen nicht zulassen, dass die Ärmsten gegeneinander ausgespielt werden.* ***[SEP]***

**1.2 Pretraining**

Originally, BERT was pretrained on a huge corpus of Wikipedia texts. In order to cope with the challenges of a directional approach while predicting the next word in a sequence, BERT uses two basic strategies during pretraining: *Masked Language Modelling* and *Next Sentence Prediction.*

**1.2.1 Masked Language Modelling (MLM)**

By masking a random word, BERT tries to predict this independent from its positioning in the sequence. As BERT is able to work in bidirectional way, that is to condition the predictions for the masked word on the co-occurred words on both sides, it is able to capture different and more flexible information about the context of a certain word. After randomly choosing 15% of the token embeddings, the 80% of them will be replaced by the “[MASK]” token, in 10% of the cases, the masked tokens are replaced by a random token (different) and in the 10% remaining cases, the masked tokens are left unchanged. Afterwards these sequences are fed into the BERT model. (Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional Transformers for language understanding.)

*[CLS] Die Ausgrenzung von* ***[MASK]*** *von der # EssenerTafel ist inakzeptabel und* ***[MASK].*** *Wir dürfen nicht zulassen, dass die* ***[MASK]*** *gegeneinander ausgespielt werden. [SEP]*

**1.2.2 Next Sentence Prediction (NSP)**

Furthermore, to capture the relationship between two sentences BERT learns to predict whether the second sentence in a pair is the follow-up one in the original document or not.In the training procedure the half of the inputs are originally a pair, while the other half consists of a random sentence from the corpus as second sentence. The goal is to make BERT distinguish between real and fake pairs.

These both strategies are applied together in order to minimize the additively combined loss function.

*Input:*

*Sentence A: [CLS] Die Ausgrenzung von [MASK] von der # EssenerTafel ist inakzeptabel und [MASK]. [SEP]*

*Sentence B: Wir dürfen nicht zulassen, dass die [MASK] gegeneinander ausgespielt werden. [SEP]*

*Label: IsNextSentence*

*Sentence A: [CLS] Die Ausgrenzung von [MASK] von der # EssenerTafel ist inakzeptabel und [MASK].*

*Sentence B: [SEP] Freue mich sehr für ihn und auf die Zusammenarbeit. [SEP]*

*Label: NotNextSentence*

**1.3 Fine-Tuning**

BERT can be used for various language tasks, for example Question Answering, Named Entity Recognition, Sequence Classification, etc. The latter one was used in this project. To train the pretrained network on a certain task, we need to fine-tune BERT on the target task and use the available (limited) labelled data. This means, we only have to exchange the output layer adapted for the target task. In our case, we used the huggingface pytorch implementation BertForSequenceClassification (<https://huggingface.co/transformers/model_doc/bert.html#tfbertforsequenceclassification>), which adds a single linear layer on top for classification. During the fine-tuning all BERT parameters are updated. (Devlin 2018) We set the following parameters recommended to use for fine-tuning by the authors: a minibatch size of 16 sequences, a global Adam learning rate of 2e-5 and 4 as number of epochs. (Devlin 2019)

**1.4 Aspect Based Sentiment Analysis**

The Aspect Based Sentiment Analysis is a more sophisticated approach than a standard text-level sentiment analysis. Apart from classifying a given text, in this case a tweet, it focuses on extracting aspects mentioned in a given text and base the classification result on it, together with the tokens contained in the text in hope to extract the most relevant information from the textual data. For our data situation, the most appropriate approach is therefore the methodology described by Hu Xu (<https://www.aclweb.org/anthology/N19-1242.pdf>)

**1.4.1 Posttraining**

The basic pretrained BERT model used in our work is the “bert-base-german-cased” model, as we deal with tweets in German language and expecting to have advantaged with the chance that letter casing will be helpful, because all nouns start with the capital letter in German. This model was originally pretrained using the latest German Wikipedia texts, news articles and Open Legal Datasets of German court decisions and citations. The dataset has a size of ca. 12 GB in total. (<https://deepset.ai/german-bert>)

(http://openlegaldata.io/research/2019/02/19/court-decision-dataset.html)

In order to improve both the domain and task knowledge the authors recommend to apply a so-called posttraining task based on contextually related data because fine-tuning BERT directly with a limited amount of labelled data may end up with domain and task challenges. The posttraining on domain knowledge is applied by using the pretrained weights as initialization and leveraging the both pretraining tasks MLM and NLP. The weights will be updated based on the sum of the losses of both tasks.

**1.4.2 Aspect Extraction (AE)**

The Aspect Extraction task is supposed to find aspects in a given text that the reviewer or in our case the user of the twitter account has commented on. The idea is to use a supervised technique and label each token from a sequence with one of these three labels: “B” – Beginning of an aspect, “I” – Inside of an aspect term and “O” – Outside of an aspect. Afterwards for each position of the sequence a dense layer and a softmax is applied to predict one of the three labels for all positions of a sequence. As shown in the original paper, the AE task requires exhaustive domain knowledge.

**1.4.3 Aspect Sentiment Classification (ASC)**

The Aspect Sentiment Classification task tries to classify the sentiment polarity of a given text, generally into positive, negative or neutral but in case of our project only in positive or negative categories. The two inputs for this task are an aspect and a review sentence or a tweet containing this aspect, whereas the aspects are either extracted automatically with the above methodology or are made available beforehand by another technique. After of application of the softmax activation and training with cross entropy loss on the polarities we get probabilities predicted for each of the sentiment categories.

**1.5 Application**

All of the models and methods were applied in Python, version 3.7.10 (https://www.python.org/downloads/release/python-3710/) using Google Colaboratory (https://colab.research.google.com/notebooks/intro.ipynb), where GPUs are available for free, which accelerated our training process.

As mentioned above, we adopted the “bert-base-german-cased” model as the basis for our experiments.

The overall application can be divided into two general parts: Sentiment Analysis (SA) and Aspect Based Sentiment Analysis (ABSA).

For the SA we have 4 variations. Apart from using the basis model by directly fine-tuning that on our train set, 972 tweets, that is using 80 % of our labelled data, we additionally enrich our training instances with 6444 Germeval data (Customer reviews about “Deutsche Bahn” - he german public train operator with about two billion passengers each year). The other 20% of the manually labelled data is used to evaluate the goodness of prediction fit. Furthermore, we develop our basic model by posttraining it on domain knowledge, namely on 29715 unlabelled and unused tweets scraped for this project (but not used for the training purposes). This model will then be fine-tuned. And finally, the fourth model implemented for SA is another pretrained German-language model named “bert-base-german-dbmdz-cased” and pretrained on a larger data source than the basic model: recent Wikipedia dump, EU Bookshop corpus, Open Subtitles, CommonCrawl, ParaCrawl and News Crawl. The dataset has a size of 16GB and 2,350,234,427 tokens, promising to have better provide even better results (https://huggingface.co/dbmdz/bert-base-german-cased)

For the ABSA we posttrained the basic model additionally on the Germeval dataset, but did not use Germeval as part of training instances, as the aspects deviate substantially from those detected in our dataset.

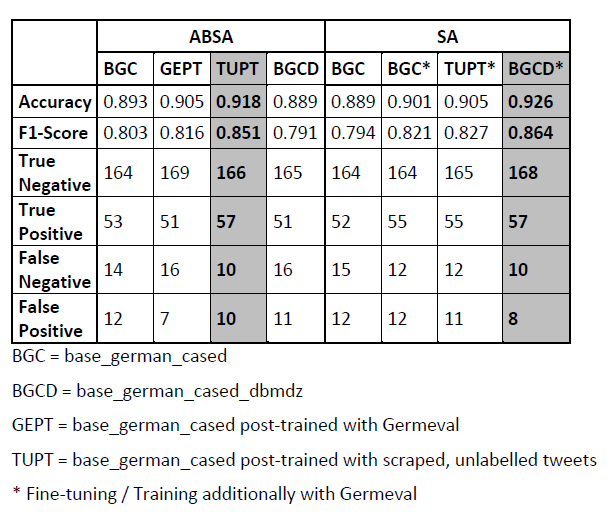
The methodology for ABSA is widely used in sentiment analysis of review texts and will be applied on tweets with political context in this project. Unfortunately, the application of AE was not successful for our project, as we could not reach any aspect detection for the tweets. There is a strong presumption that this happens because of the lack of clear contextual and semantic delimitation aspect containing words. We therefore see good opportunities to use this implementation for other use cases and provide code background for further research but this will not be discussed any further as part of this report. (s. el. Anhang) Therefore for the ABSA task we used the manually assigned aspects for both training and evaluation procedures.

**1.6 Results**

With the aim of evaluating how good each of the models in both parts predict, we consider the leave out sample of 243 test (manually labelled) instances.

The following metrics are used: the accuracy score, the F1-Score and the elements of confusion matrix, namely the amount of True Negative, True Positive, False Negative and False Positive predictions.

The table below shows the results of in total eight models implemented for both analysis approaches. The columns marked in grey represent the best result of each approach.



For the SA the best result is reached with the base\_german\_cased\_dbmdz model with additionally fine-tuning on the 80% of our scraped tweets and the Germeval dataset with 0.93 accuracy score, 168 True Negative and 57 True Positive predictions.

If we consider the aspects detected in the tweets on top of that the best result will be achieved with the base\_german\_cased model, posttrained with unlabelled scraped tweets. Here we have an accuracy score of 0.92, 166 True Negative and 57 True Positive predictions.

Noticeable is that the additional consideration of aspect terms leads to only a minimal increase of prediction goodness for almost all of the models (with exception of BGCD).

However, we can state that the application of BERT and its variants results in satisfactory outcome in classifying tweets into positive and negative categories for both Sentiment Analysis and Aspect Based Sentiment Analysis tasks.