GottBERT: a pure German Language Model

Raphael Scheible[1], **Johann Frei**[2], **Fabian Thomczyk**[3], **Henry He**[4], **Patric Tippmann**[5,6], **Jochen Knaus**[5], **Victor Jaravine**[7], **Frank Kramer**[2] and **Martin Boeker**[1]

[1] Institute for AI and Informatics in Medicine, University Hospital rechts der Isar, Technical University Munich
[2] IT-Infrastructure for Translational Medical Research, Faculty of Applied Computer Science, University of Augsburg
[3] Data Integration Center, Faculty of Medicine, University of Freiburg
[4] School of Computation, Information and Technology, Technical University Munich
[5] Institute of Medical Biometry and Statistics, Medical Center, Faculty of Medicine, University of Freiburg
[6] Freiburg Center for Data Analysis and Modeling, University of Freiburg
[7] Hengrui Europe Biosciences, Zurich

Correspondence: raphael.scheible@tum.de

Abstract

Pre-trained language models have significantly advanced natural language processing (NLP), especially with the introduction of BERT and its optimized version, RoBERTa. While initial research focused on English, single-language models can be advantageous compared to multilingual ones in terms of pre-training effort, overall resource efficiency or downstream task performance. Despite the growing popularity of prompt-based LLMs, more compute-efficient BERT-like models remain highly relevant. In this work, we present the first German single-language RoBERTa model, GottBERT, pre-trained exclusively on the German portion of the OSCAR dataset. Additionally, we investigated the impact of filtering the OSCAR corpus. GottBERT was pre-trained using fairseq and standard hyperparameters. We evaluated its performance on two Named Entity Recognition (NER) tasks (Conll 2003 and GermEval 2014) and three text classification tasks (GermEval 2018 fine and coarse, and 10kGNAD) against existing German BERT models and two multilingual models. Performance was measured using the F_1 score and accuracy. The GottBERT base and large models showed competitive performance, with GottBERT leading among the base models in 4 of 6 tasks. Contrary to our expectation, the applied filtering did not significantly affect the results. To support the German NLP research community, we are releasing the GottBERT models under the MIT license.

1 Introduction

The computation of contextual pre-trained word representations is the foundation of neural language modeling (LM) in natural language processing (NLP). The field of NLP experienced remarkable progress by the use of transformer-based approaches (Vaswani et al., 2017). Especially Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) impacted the field which subsequently was robustly optimized to RoBERTa (Liu et al., 2019). These transformer-based approaches rely on large-scale pre-trained language models, which are subsequently fine-tuned through supervised training on specific downstream tasks, leveraging the context representations learned from the generic domain to achieve superior performance compared to training from scratch, a process known as transfer learning. On the other hand, the computation of the language model is performed self-supervised. Large text blobs are required for training and strong hardware such as hundreds of Graphics Processing Units (GPU) (Martin et al., 2020) or Tensor Processing Units (TPU) (You et al., 2020). Initially, most of the research took place in English followed by multilingual approaches (Conneau et al., 2019; Conneau and Lample, 2019). Although, multilingual approaches were trained on large texts of many languages, they can be outperformed by single language models (de Vries et al., 2019; Martin et al., 2020; Le et al., 2020; Delobelle et al., 2020). Additionally, a single language model requires fewer computational resources and a smaller dataset compared to the vast and varied data needed for multilingual models. Single language models trained with the Open Super-large Crawled ALMAnaCH coRpus (OSCAR) (Ortiz Suárez et al., 2020) showed good performance due to the size and variance of the OSCAR corpus (Martin et al., 2020; Delobelle et al.,

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 21237–21250

2020). The focus of pre-training language models has shifted towards scaling up transformer-based large language models (LLMs) (Touvron et al., 2023a,b; Liu et al., 2023; Jiang et al., 2023). These models, like Llama 3[1], are vastly larger than the aforementioned models (Touvron et al., 2023a,b). Despite the advantages of LLMs, such as gradient-free prompting, smaller models remain valuable for their efficiency and practical deployment. For these reasons, we pre-trained the first German RoBERTa single language models with the German portion of the first published deduplicated version of OSCAR - the German OSCAR text trained BERT (GottBERT). Inspired by FlauBERT (Le et al., 2020), we also trained models with a filtered version of OSCAR. In an evaluation we compared the performance of all models on the two named entity recognition tasks Conll 2003 and GermEval 2014, NLI as well as on the text classification tasks GermEval 2018 and GNAD with existing German single language BERT and models and two multilingual models.

Our contributions can be summarized as follows:

- We introduced a filtering method specifically applicable to German texts which we applied to the first version of the German portion of the OSCAR corpus.
- We pre-trained single language RoBERTa models specifically for the German language based on the filtered and original OSCAR corpus. These models are publicly available under the MIT open-source license.
- We evaluated the models on five downstream tasks (3 classification, 2 NER and NLI). Further, we demonstrated the effects of training the model with the filtered corpus.

2 Related Work

Most recently, transformer-based models widely impacted the field of NLP. From neural translation (Ott et al., 2018; Ng et al., 2019) to generative language models starting with GPT-2 (Radford et al., 2019), remarkable performance gains were achieved. With BERT, an approach to facilitate pre-trained transformer-based models was introduced. Fine-tuned on downstream tasks, BERT-based approaches improved the performance of several NLP tasks (Devlin et al., 2019; Liu et al., 2019). However, BERT models were first released as single-language models in English based on 16GB of raw text and as the multilingual model mBERT based on Wikipedia in about 100 languages (Devlin, 2018). These models were followed by single-language models for several languages: Bertje (de Vries et al., 2019) for Dutch, FinBERT (Virtanen et al., 2019) for Finish, GermanBERT [2] and a German BERT from the MDZ Digital Library team at the Bavarian State Library to which we refer to as dbmz BERT in this paper[3]. GermanBERT was trained using 12GB of raw text data basing on the German Wikipedia (6GB), the OpenLegalData dump (2.4GB) and news articles (3.6GB). dbmz BERT used as source data a German Wikipedia dump, EU Bookshop corpus, Open Subtitles, CommonCrawl, ParaCrawl and News Crawl which sums up to a dataset of 16GB. With the release of RoBERTa a new standard for raw text size was set as it was trained on 160GB of raw English text. Further, RoBERTa enhances the original BERT approach by removing segment embeddings, next sentence prediction and improved hyperparameters. Additionally, instead of using wordpiece (Schuster and Nakajima, 2012) tokenization, RoBERTa utilizes GPT2's byte pair encoding (BPE) (Radford et al., 2019) with the benefit that language-specific tokenizers are not required. Other than mBERT, the multilingual XLM-RoBERTa (Conneau et al., 2019) was trained on 2.5TB of filtered CommonCrawl data. CamemBERT is a French RoBERTa model that was trained on the OSCAR and uses sentencepiece (Kudo and Richardson, 2018) BPE. Further, they pre-trained a model with 4GB of the French OSCAR portion and another model with 4GB of the French Wikipedia. The comparison of these models using downstream tasks shows that high text variance leads to better results. UmBERTo[4] is an Italian RoBERTa model, similarly designed as CamemBERT. RobBERT, the Dutch single language RoBERTa, was trained on 39GB of the Dutch portion of the OSCAR and outperformed Bertje. A more recent version of RobBert showed the performance gains of language specific BPE compared to the English based GPT2 BPE in downstream tasks. Susequently, FlauBERT (Le et al., 2020) for French was released trained on 71GB data. They cleaned a 270GB corpus of mixed sources by filtering out meaningless con-

[1] https://github.com/meta-llama/llama3

[2] https://deepset.ai/german-bert

[3] https://github.com/dbmdz/berts#german-bert

[4] https://github.com/musixmatchresearch/umberto

tent and Unicode-normalization. Data was pre-tokenized by moses (Koehn et al., 2007) and encoded by fastBPE[5] which is an implementation of Sennrich et al. (2016). After the publication of RoBERTa, Google released ELECTRA (Clark et al., 2020) which denoted an improvement to the BERT architecture. Based on these developments, further German language models were then published: GBERT and GELECTRA (Chan et al., 2020). These models were trained on the German portion on OSCAR (145GB) besides three other datasets (18.4GB) and outperformed previously released German single language models.

3 Methodology

Following the approach of utilizing the OSCAR, we computed the German OSCAR text trained BERT (GottBERT). However, the drawback of BERT approaches is the computational power requirement. Multiple GPUs or TPUs were used for pre-training. All previously listed RoBERTa-based models were computed on GPUs whereas GottBERT is the first published RoBERTa model pre-trained on TPUs.

Training Data

The GottBERT model is trained on the German portion of the OSCAR, a large multilingual text corpus extracted from Common Crawl. The German data portion of the first published deduplicated version of the OSCAR measures 145GB of text containing approximately 21.5 billion words in approximately 459 million documents (one document per line).

Filtering OSCAR

While screening the German OSCAR portion, some issues attracted our attention:

1. erroneous umlauts
2. meaningless documents such as spam, e.g. lists of words
3. non-German documents

We were able to trace back the cause of wrong umlauts to decoding errors. According to our findings, when an umlaut is considered to be a non-UTF8 encoding, but actually is already UTF8, wrong characters are generated (see Table 1). In other cases, where the encoding wasn't reproducible, the sign `&#65533;`, is shown. Consequently, lines with at least one `&#65533;` were removed. To the rest of documents, we corrected the encoding by applying clean-text[6]. The tool was further configured to remove phone numbers, email addresses, URLs and emojis. Also only documents with a length of at least 40 characters were considered. Secondly, we applied a language detection algorithm which especially filtered lines belonging to ASCII arts (see Apendix A). Due to the corpus size, an efficient Rust implementation[7] of a language detection based on n-gram based text categorization was used (Cavnar and Trenkle, 1994).

Finally, also due to the corpus size, we trained and applied a single-class SVM (Schölkopf et al., 1999) which was trained to filter meaningless documents. In this respect, a special feature of the German language is the capitalization of nouns. In the 17th and 18th century there was a trend in the English and Swedish languages to write nouns with an initial capital (Crystal and Crystal, 2003; Solling, 2009). Dutch had this rule until a spelling reform in 1948. German kept this special rule, although undergoing several orthography reforms in the 20th century. However, Germans sometimes tend to neglect this rule, especially in social media. Therefore, social media platforms might not be a good source for texts for research requiring good quality of orthography, although these texts lead to admirable results in the generative model GPT-2 (Radford et al., 2019). In the German OSCAR portion, it was noticeable that documents with little meaning were often written completely in capital letters, had many punctuation marks in relation to words, had a lot of nouns or no stop words at all. Based on this knowledge, the following ratios were computed for each document D consisting of tokens $t_0, t_1, ..., t_{n-1}, t_n$:

- stopword ratio:

$$r_s = \frac{\sum_i^n \sum_{t_s \in S} [t_i = t_s]}{n},$$

  where S is the set of stop word tokens for which we used nltk's stopword list.

- punctuation ratio:

$$r_p = \frac{|\{\, t_i \,|\, t_i \notin W(D) \,\}|}{n},$$

[5] https://github.com/glample/fastBPE

[6] https://github.com/jfilter/clean-text

[7] https://github.com/greyblake/whatlang-rs

where $W(D)$ are the word tokens $w_0, w_1, .., w_{m-1}, w_m$ of D.

- unique words ratio:

$$r_u = \frac{|\{\, w_i \,|\, \forall w_j \in W(D) : w_i \neq w_j \}|}{n},$$

- upper token ratio:

$$r_{up} = \frac{\sum_{w_i \in W(D)} c(w_i)}{n},$$

where

$$c(w) = \begin{cases} 1 & \text{if w is capital} \\ 0 & \text{otherwise} \end{cases}$$

To train the single class SVM, 12k documents were used for the unsupervised training. The performance was measured based on 1750 documents annotated documents consisting of two classes: 334 dirty (15.5%) and 1466 clean (81.5%). Documents classified as dirty were content of weak meaning, including lists of words, source code, documents mainly using ASCII signs and documents lacking of spaces or having wrong spacing between words. The single hyperparameter nu was optimized by a grid search. The best performing SVM had a weighted F_1-score of 0.8578 and 0.5663 as Matthews correlation coefficient (Chicco and Jurman, 2020; Chicco et al., 2021). After filtering, the corpus measured 121GB of text containing approximately 18.1 billion words in approximately 382 million documents (one document per line).

Pre-processing

Originally, RoBERTa uses GPT-2 (Radford et al., 2019) byte pair encoding to segment the input into subword units. Therefore, no pre-tokenization is required and thus no language-specific tokenizer as e.g. moses (Koehn et al., 2007) must be used. Its original vocabulary was computed on English data. For GottBERT we computed a vocabulary of 52k subword tokens based on 40 GB randomly sampled documents of the German OSCAR portion. Compared to the original GPT-2 tokenizer, which was trained on English data, this leads to a 40% smaller size of the binary data which are fed into fairseq (Ott et al., 2019). Furthermore, according to Delobelle et al. (2020), it leads to a performance increase.

Pre-training

Using fairseq, we pre-trained the GottBERT_{base} model using the unfiltered OSCAR on a 256 core TPUv3 pod. The remaining GottBERT models were computed on a 128 core TPUv4 (Jouppi et al., 2023) pod. We trained the models with RoBERTa base architecture in 100k update steps using a batch size of 8k. A 10k iteration warmup of the learning rate to a peak of 0.0004 was applied, from which the learning rate polynomially decayed to zero. The models with RoBERTa large architecture were trained with the same properties but a peak learning rate of 0.00015. After training on both the filtered and unfiltered OSCAR datasets, we developed four models: GottBERT_{base} and GottBERT_{large} using the unfiltered as well as $^{\text{f}}\text{GottBERT}_{base}$ and $^{\text{f}}\text{GottBERT}_{large}$ using the filtered dataset. Further, we evaluated each epoch and saved its checkpoint, potentially leading to multiple checkpoints per model setup, namely best and last. The latter ones are indicated with a †, e.g. $\text{GottBERT}^{\dagger}_{base}$. The base models took ca. 1.2 days computation time, while the large ones computed ca. 5.7 days.

Downstream Tasks

Based on the pre-trained BERT models, several downstream tasks were trained. The training was conducted using the scripts provided by Huggingface (Wolf et al., 2019). Hyperparameter optimization was performed through a grid search focusing on batch size and learning rate. We trained the downstream tasks NER and CLS with a maximum of 30 epochs.

For natural language inference (NLI), we utilized the hyperparameters specified by Facebook (originally implemented in Fairseq), adopting them to the extent they were available within the Huggingface framework. These tasks were trained with a maximum of 10 epochs.

In order to evaluate the performance, each downstream task ran 24 times using different batch sizes and learning rates. To determine the best checkpoint after training, we select the checkpoint that yields the best F_1 scores (accuracy for NLI) on the evaluation set. The score is the best of 24 runs of the respective experiment of each trained model. The best score selection is based on the validation set. In terms of performance, our models were compared with six other models listed in Table 2.

NLI NLI entails predicting whether a hypothesis sentence is entailed by, neutral towards or contra-

Special Character	ä	ü	ö	ß	Ä	Ü	Ö
ISO-8859-1	Ã¤	Ã¼	Ã¶	Ã[U+009F]	Ã[U+0084]	Ã[U+009C]	Ã[U+0096]
ISO-8859-2	Ă¤	Ăź	Ăś	Ă[U+009F]	Ă[U+0084]	Ă[U+009C]	Ă[U+0096]
ISO-8859-4	Ã¤	Ã[U+0167]	Ãļ	Ã[U+009F]	Ã[U+0084]	Ã[U+009C]	Ã[U+0096]
ISO-8859-9	Ã¤	Ã¼	Ã¶	Ã[U+009F]	Ã[U+0084]	Ã[U+009C]	Ã[U+0096]
ISO-8859-10	ÃĪ	Ãž	Ãķ	Ã[U+009F]	Ã[U+0084]	Ã[U+009C]	Ã[U+0096]
ISO-8859-16	Ă€	ĂŒ	Ă¶	Ă[U+009F]	Ă[U+0084]	Ă[U+009C]	Ă[U+0096]
Windows-1250	Ă¤	ĂĽ	Ă¶	Ăź	Ă„	Ăś	Ă–
Windows-1252	Ã¤	Ã¼	Ã¶	ÃŸ	Ã„	Ãœ	Ã–

Table 1: This table shows the result of wrong encoding of German umlauts. Artifacts occur whenever a file is expected to be encoded by an appropriate encoding, but truly is UTF-8 encoded.

Model	**Type**	**#lang**	**Data Size**	**Data Source**
GottBERT	RoBERTa	1	145GB	OSCAR
[f]GottBERT	RoBERTa	1	121GB	filtered OSCAR
GBERT	BERT	1	163.4	OSCAR, OPUS, Wikipedia, OpenLegalData
GELECTRA	ELECTRA	1	163.4	OSCAR, OPUS, Wikipedia, OpenLegalData
dbmz BERT	BERT	1	16GB	Wikipedia, EU Bookshop corpus[8], Open Subtitles, Common-,Para-,NewsCrawl
$\text{mBERT}_{\text{cased}}$	BERT	104	unknown	Wikipedia
GermanBERT	BERT	1	12GB	news articles, Open Legal Data[9], Wikipedia
XLM RoBERTa	RoBERTa	100	2.5TB (66.6GB German)	CommonCrawl, Wikipedia

Table 2: This table shows the models, we used in our experiments. Additional information about the pre-training and architecture is listed. #lang is the number of languages. Unfortunately, for mBERT we did not find any estimate about the data size.

dicts a premise sentence. We assessed our model on NLI using the German portion of the XNLI dataset (Conneau et al., 2018). The XNLI dataset is an extension of the Multi-Genre NLI (MultiNLI) corpus Williams et al. (2018), expanded to 15 languages by manually translating the validation and test sets into each language. For languages other than English, the training set is machine translated. The dataset includes 122k training examples, 2490 development examples, and 5010 test examples for each language. Typically, NLI performance is measured using accuracy.

Named Entity Recognition We evaluated GottBERT on two NER tasks. One was the German part of CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003). It contains three main entity classes and one for other miscellaneous entities. As measurement we used the *harmonic mean of precision and recall* F_1. The second NER task was GermEval 2014 (Benikova et al., 2014). It extends the CoNLL 2003 shared task by fine-grained labels and embedded markables. Fine-grained labels allow the indication of NER subtypes common in German, namely derivations and parts: e.g. "Mann" $\rightarrow$ "männlich" and "Mann" $\rightarrow$ "mannhaft". In order to recognize nested NEs embedded markables are required. Specifically, this was realized by annotating main classes as well as two levels of subclasses. Performance was measured by the use of an adapted F_1 evaluation metric Benikova et al. (2014), which considers the equality of labels and spans (text passages) and additionally levels in the class hierarchy.

Text Classification GermEval task 2018 (Risch et al., 2018) is a text classification task that contains two subtasks of different granularity: the coarse-

grained binary classification of German tweets and fine-grained classification of the same tweets into four different classes. As this datasets does not provide a pre-defined validation set, we used 54% of the original training set for training, 6% for validation and 40% for test. With this split decision, we sticked to Chan et al. (2020). Based on the One Million Posts Corpus (Schabus et al., 2017), the 10k German News Articles Dataset (10kGNAD) topic classification benchmark[10] was created. The dataset contains approximately 10k news articles from an Austrian newspaper which are to be classified into 9 categories. Usually, 10kGNAD does not provide a pre-defined split. However, the version we used provides a split, using 90% of the original set for training and 10% for test. We split 10% from the training set for validation. For evaluation of both tasks we computed the mean of the F_1-scores of each class/category.

4 Results

As GottBERT_{large} was the same checkpoint for last and best, we ended up with 7 GottBERT checkpoints. GottBERT_{base} was saved after 91848 training steps (12 epochs). The filtered models, both $^{\text{f}}\text{GottBERT}_{large}$ and $^{\text{f}}\text{GottBERT}_{base}$, were saved also saved after they trained 94530 steps (15 epochs). For these models, this was approximately 1 epoch earlier then the full training steps. The dirty models trained up to 13.07 epochs and the filtered ones up to 15.87 epochs. The results of all the downstream tasks are listed in Table 3.

NLI Among the large models, GottBERT_{large} achieved an accuracy of 82.46%, while $^{\text{f}}\text{GottBERT}_{large}$ and $^{\text{f}}\text{GottBERT}^{\dagger}_{large}$ slightly improved on this with accuracies of 83.31% and 82.79%, respectively. These results position the GottBERT models as strong contenders, though they were outperformed by $\text{GELECTRA}_{\text{large}}$, which achieved the highest accuracy of 86.33%. $\text{GBERT}_{\text{large}}$ also performed well with an accuracy of 84.21%, followed closely by $\text{XLM-R}_{\text{large}}$ with 84.07%.

For the base models, GottBERT_{base} and $\text{GottBERT}^{\dagger}_{base}$ achieved accuracies of 80.82% and 81.04%, respectively, demonstrating competitive performance. $^{\text{f}}\text{GottBERT}_{base}$ and $^{\text{f}}\text{GottBERT}^{\dagger}_{base}$ had similar accuracies of 80.56% and 80.74%, respectively. Among the base models, $\text{GELECTRA}_{\text{base}}$ outperformed the others with an accuracy of 81.70%. $\text{GBERT}_{\text{base}}$ scored slightly lower with 80.06. Other models like GermanBERT and $\text{XLM-R}_{\text{base}}$ achieved 78.16% and 79.76%, respectively, while dbmdzBERT and mBERT had the lowest accuracies at 68.12% and 77.03%.

Overall, the results indicate that while the GottBERT models exhibit strong performance in the NLI task, GELECTRA models generally achieved the highest accuracies in both the base and large categories.

Named Entity Recognition For the NER tasks, the base versions of the GottBERT models showed competitive performance with F1 scores around 87.50% on the GermEval 2014 dataset and around 86.10% on the CoNLL dataset. The large versions of these models improved upon these scores, with $^{\text{f}}\text{GottBERT}_{last}$ achieving an F1 score of 88.27% on GermEval 2014 and 86.78% on CoNLL. However, among the large models, XLM-R achieved the highest F1 score of 88.83 on the GermEval 2014 dataset, whereas $\text{GBERT}_{\text{large}}$ performed the best on the CoNLL dataset with an F1 score of 87.19%. Overall, the large GottBERT models demonstrated robust performance across both datasets, validating their effectiveness for the tasks at hand. Among the base architecture the GottBERT models took the lead.

Text Classification For GermEval 2018, the large GottBERT models showed again competitive performance. The GottBERT_{large} and $^{\text{f}}\text{GottBERT}_{large}$ models achieved overall F1 scores of around 79.3 for coarse-grained predictions, with minimal differences in fine-grained scores around 54.7. $^{\text{f}}\text{GottBERT}^{\dagger}_{large}$ had slightly lower performance in coarse predictions but was consistent in fine-grained predictions. In comparison, $\text{GELECTRA}_{\text{large}}$ outperformed all large models in coarse-grained predictions with an F1 score of 81.28, and also showed strong fine-grained performance with an F1 score of 56.17. $\text{GBERT}_{\text{large}}$ followed closely with 80.84 in coarse-grained predictions and led in fine-grained predictions with 57.37. $\text{XLM-R}_{\text{large}}$ scored slightly lower than the GottBERT models, with 79.05 and 55.06 in coarse and fine-grained predictions, respectively.

Among the base models, GottBERT_{base} scored 78.17 for coarse-grained predictions and 53.30 for fine-grained predictions. $\text{GottBERT}^{\dagger}_{base}$ showed similar performance in coarse-grained prediction with an F1 score of 78.18 and a fine-grained F1

[10] https://huggingface.co/datasets/community-datasets/gnad10

Model	XNLI	GermEval 2014	CoNLL 03	GermEval 2018 coarse	GermEval 2018 fine	10kGNAD
GottBERT_{base}	80.82	87.55	85.93	78.17	53.30	89.64
$\text{GottBERT}^{\dagger}_{base}$	81.04	87.48	85.61	78.18	**53.92**	90.27
$^{\text{f}}\text{GottBERT}_{base}$	80.56	87.57	**86.14**	**78.65**	52.82	89.79
$^{\text{f}}\text{GottBERT}^{\dagger}_{base}$	80.74	**87.59**	85.66	78.08	52.39	89.92
$\text{GELECTRA}_{\text{base}}$	**81.70**	86.91	85.37	77.26	50.07	89.02
$\text{GBERT}_{\text{base}}$	80.06	87.24	85.16	77.37	51.51	90.30
dbmdzBERT	68.12	86.82	85.15	77.46	52.07	**90.34**
GermanBERT	78.16	86.53	83.87	74.81	47.78	90.18
$\text{XLM-R}_{\text{base}}$	79.76	86.14	84.46	77.13	50.54	89.81
mBERT	77.03	86.67	83.18	73.54	48.32	88.90
GottBERT_{large}	82.46	88.20	86.78	79.40	54.61	90.24
$^{\text{f}}\text{GottBERT}_{large}$	83.31	88.13	86.30	79.32	54.70	90.31
$^{\text{f}}\text{GottBERT}^{\dagger}_{large}$	82.79	88.27	86.28	78.96	54.72	90.17
$\text{GELECTRA}_{\text{large}}$	**86.33**	88.72	86.78	**81.28**	56.17	**90.97**
$\text{GBERT}_{\text{large}}$	84.21	88.72	**87.19**	80.84	**57.37**	90.74
$\text{XLM-R}_{\text{large}}$	84.07	**88.83**	86.54	79.05	55.06	90.17

Table 3: All the results of the experiments are shown in percent. They are all based on the test set and the best score out of 24 runs (selection based on validation set). While NLI is measured by accuracy, all the other metrics are F_1 measures. Best score in bold and second underlined, for large and base models respectively.

score of 53.92. $^{\text{f}}\text{GottBERT}_{base}$ achieved the highest coarse-grained F1 score of 78.65 but had a lower fine-grained score of 52.82. $\text{GELECTRA}_{\text{base}}$ scored 77.26 and 50.07 for coarse and fine-grained predictions and therefore scored close to $\text{XLM-R}_{\text{base}}$. $\text{GBERT}_{\text{base}}$ and dbmdzBERT demonstrated moderate performance, while GermanBERT and mBERT had the lowest scores.

The evaluation of GottBERT models on the 10kGNAD dataset demonstrated their strong performance in German news classification tasks. For the large models, GottBERT_{large} achieved an accuracy of 90.24, while $^{\text{f}}\text{GottBERT}_{large}$ and $^{\text{f}}\text{GottBERT}^{\dagger}_{large}$ scored 90.31 and 90.17, respectively. Among the competing models, $\text{GELECTRA}_{\text{large}}$ outperformed all with an accuracy of 90.97, followed by $\text{GBERT}_{\text{large}}$ at 90.74, and $\text{XLM-R}_{\text{large}}$ matching $^{\text{f}}\text{GottBERT}^{\dagger}_{large}$ at 90.17.

For the base models, $\text{GottBERT}^{\dagger}_{base}$ excelled with an accuracy of 90.27, while GottBERT_{base} and $^{\text{f}}\text{GottBERT}_{base}$ achieved 89.64 and 89.79, respectively. $^{\text{f}}\text{GottBERT}^{\dagger}_{base}$ performed well with an accuracy of 89.92. Notably, dbmdzBERT scored the highest among the base models with 90.34, followed closely by $\text{GBERT}_{\text{base}}$ at 90.30. GermanBERT, $\text{XLM-R}_{\text{base}}$, and mBERT also showed competitive accuracies ranging from 88.90 to 90.18.

Overall, GottBERT models demonstrate strong and consistent performance across the classification tasks, highlighting their robustness and effectiveness.

5 Discussion

In this study, we successfully trained and evaluated GottBERT models on two versions of the OSCAR corpus. Noteworthily, Scheible et al. (2020) published $\text{GottBERT}^{\dagger}_{base}$ as preliminary work. Since its release, the model has been utilized for various purposes in several related works showing its relevance. It has served as a baseline model in research studies (Scherrmann, 2023; Bressem et al., 2024; Lentzen et al., 2022). Beyond that, in the field of neural machine translation (NMT), researchers have used contextualized embeddings from pre-trained models including GottBERT(Xu et al., 2021). Additionally, the model has been applied to named entity recognition (NER) tasks in the medical field, using both translated (Frei et al., 2022) and synthetic (Frei and Kramer, 2023) NER data annotated with medical entities through a fine-tuned version of GottBERT. Furthermore, a specialized version of the model known as BioGottBERT has been developed specifically for the medical domain (Lentzen et al., 2022).

TPU training generally does not permit dynamic memory allocation, as TPUs are designed for ef-

ficient, high-throughput computation with fixed memory allocation. As a result, the corpus was processed as a single stream rather without considering document boundaries, unlike RoBERTa training on GPUs. Additionally, due to limitations in the fairseq implementation we used, we conducted the computations in 32-bit mode since 16-bit was neither properly implemented nor tested, leading to increased memory usage and hence more computation time required. Also, we used more conservative learning rates than the ones recommended by the fairseq documentation for pre-training on GPU.

Dataset annotation is expensive, as it is usually performed by multiple annotators. As the one-class SVD is trained unsupervised and annotations were only used to estimate its performance and to find a good nu, the dataset was only annotated by one person of our team. The MCC of 0.5663 suggests the model has a moderate ability to make accurate binary predictions overall, balancing true and false positives and negatives. The F1-score of 0.8578 indicates the model is performing well in terms of precision and recall for the positive class. These metrics together imply that while the model is quite good at correctly identifying positive cases and maintaining a balance between precision and recall, there is still room for improvement in making more accurate overall predictions as reflected in the moderate MCC score. Possibly a better approach would have been possible with a more complex model, the computational cost as well as the annotation efforts would have been much more expensive. However, the use of language properties on a syntactical level denoted an efficient and creative approach that could be carried out with fair effort, including in terms of computational costs.

Our results do not provide a clear conclusion regarding the impact of data cleaning on the resulting model. A ranking of all base models by their position, taking into account only the number of first and second place models, shows that $^{\text{f}}\text{GottBERT}_{base}$ is on top. However, when considering the GottBERT models in isolation as a subgroup, the $\text{GottBERT}^{\dagger}_{base}$ model stands out as the top performer among the base models. Conversely, when evaluating the large models all GottBERT models were outperformed by the competitors. In this global comparison, the unfiltered model GottBERT_{large} emerges as the best performer of all the GottBERT models winning one second place in CoNLL03, while $^{\text{f}}\text{GottBERT}_{large}$ emerges as the superior performer in the isolated comparison. We anticipated a more definitive outcome, particularly since the filtered models were pre-trained for an additional epoch due to the smaller corpus size. The importance of hyperparameters in model performance is well-documented, even considering random seeds as shown by Dodge et al. (2020). This suggests that our chosen hyperparameters could be extended even more to find better ones. Nevertheless, clear differences should have been already pointed out within our experimental setup. However, we did not see any great benefit, especially considering the high cost of cleaning a data set in this way.

Potentially, the data cleaning process might have inadvertently removed important variance from the corpus. According to the Martin et al. (2020), a corpus with greater variance generally leads to better performance compared to a homogeneous one. Therefore, we suggest creating a corpus with more variance. In our case, incorporating additional corpora such as OPUS, Wikipedia, and OpenLegalData could have been beneficial. Moreover, whole word masking (WWM) leads to better models (Martin et al., 2020; Chan et al., 2020). Finally, for RoBERTa models, the size of the vocabulary also impacts performance, as investigated by Toraman et al. (2023). According to their findings, our vocabulary size wasn't a bad choice.

Finally, potential risks include bias and fairness issues, leading to unfair outcomes. Data privacy concerns exist, with the model potentially revealing sensitive information. This affects especially the corpus used and the filtered version of it, as the filtering did not operate on a semantical but on a syntactical level. Misuse could result in harmful content, like misinformation or spam. Over-reliance without human oversight might cause critical errors, especially in healthcare or finance. The environmental impact of training such models is considerable due to high energy consumption. It is also vulnerable to adversarial attacks.

6 Conclusion

In this work we present the German single language RoBERTa based model GottBERT in two versions computed on a corpus with 145GB and a filtered version with 121GB plain text with both base and large RoBERTa architecture. GottBERT is the first German single language RoBERTa based model. In our experiments, we were able to show that the base models lead 4 of 6 tasks. However,

this did not apply to the large models. The comparison of the pre-training with filtered and raw corpus did not show a clear result as anticipated. We therefore suggest considering other measures, such as increasing variance by using many corpora and using WWM. We release all GottBERT models in Huggingface and fairseq format to the community under the MIT license.

Acknowledgments

This work was supported by the German Ministry for Education and Research (BMBF FKZ 01ZZ1801B, 01ZZ1804A, 01KX2121, and 01ZZ2304A) and supported with Cloud TPUs from Google's TPU Research Cloud (TRC). We would like to thank Ian Graham for constructive criticism of the manuscript and Louis Martin for the helping email contact. A special thanks goes to Myle Ott, who implemented the TPU feature in fairseq and intensively supported us to get our computation run. Finally, we would like to recognizably thank the people behind the scenes who essentially made this work possible: Frank Werner, Georg Koch and Friedlinde Bühler of the IMBI in Freiburg, Andreas Enterrottacher of the AIIM team in Munich, Philipp Munz and Christian Wiedemann of Wabion GmbH, Carsten Peters of Digital Schooling UG and last but not least Nora Limbourg the Google Cloud Customer Engineer assigned to us.

Limitations

Several limitations need to be acknowledged in this study. First, the cleaning algorithm used was simple yet fast, but it could still be enhanced. Improved cleaning methods might lead to a clearer picture.

Further, the results obtained are specific to the first version of the OSCAR corpus, and this perspective may not generalize to other corpora or different languages. The performance of the models might vary significantly when applied to different datasets. Our evaluation tasks have already indicated this. The models may struggle with cultural nuances and dialects within German. While GottBERT performs well on the used benchmarks, it may not adapt to evolving language use without further training.

Lastly, due to limited resources, we did not experiment with various learning rates (LR) for the pre-training of the large models. We chose a conservative peak LR of 0.00015, assuming that other learning rates could potentially lead to significantly better performance. However, given that large models require ca. 4.75 times more computation time, it was not feasible for us to explore different learning rates extensively.

References

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. 2014. GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, pages 104–112.

Keno K. Bressem, Jens-Michalis Papaioannou, Paul Grundmann, Florian Borchert, Lisa C. Adams, Leonhard Liu, Felix Busch, Lina Xu, Jan P. Loyen, Stefan M. Niehues, Moritz Augustin, Lennart Grosser, Marcus R. Makowski, Hugo J.W.L. Aerts, and Alexander Löser. 2024. medbert.de: A comprehensive german bert model for the medical domain. *Expert Systems with Applications*, 237:121598.

William B. Cavnar and John M. Trenkle. 1994. N-Gram-Based Text Categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6.

Davide Chicco, Matthijs J. Warrens, and Giuseppe Jurman. 2021. The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment. *IEEE Access*, 9:78368–78381. Conference Name: IEEE Access.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *Preprint*, arXiv:2003.10555.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

David Crystal and Honorary Professor of Linguistics David Crystal. 2003. *The Cambridge Encyclopedia of the English Language*. Cambridge University Press.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv:1912.09582 [cs]*. ArXiv: 1912.09582.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. *arXiv:2001.06286 [cs]*. ArXiv: 2001.06286.

Jacob Devlin. 2018. Multilingual BERT Readme Document.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv:2002.06305 [cs]*. ArXiv: 2002.06305.

Johann Frei, Ludwig Frei-Stuber, and Frank Kramer. 2022. Gernermed++: Transfer learning in german medical nlp. *Preprint*, arXiv:2206.14504.

Johann Frei and Frank Kramer. 2023. Annotated dataset creation through large language models for non-english medical nlp. *Journal of Biomedical Informatics*, 145:104478.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Norman P. Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Cliff Young, Xiang Zhou, Zongwei Zhou, and David Patterson. 2023. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. *Preprint*, arxiv:2304.01433 [cs].

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised Language Model Pre-training for French. *arXiv:1912.05372 [cs]*. ArXiv: 1912.05372.

Manuel Lentzen, Sumit Madan, Vanessa Lage-Rupprecht, Lisa Kühnel, Juliane Fluck, Marc Jacobs, Mirja Mittermaier, Martin Witzenrath, Peter Brunecker, Martin Hofmann-Apitius, Joachim Weber, and Holger Fröhlich. 2022. Critical assessment of transformer-based AI models for German clinical notes. *JAMIA Open*, 5(4):ooac087.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook

FAIR's WMT19 News Translation Task Submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. *arXiv:1904.01038 [cs]*. ArXiv: 1904.01038.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. *arXiv:1806.00187 [cs]*. ArXiv: 1806.00187.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Julian Risch, Eva Krebs, Alexander Löser, Alexander Riese, and Ralf Krestel. 2018. Fine-Grained Classification of Offensive Language. In *Proceedings of GermEval 2018 (co-located with KONVENS)*, pages 38–44.

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One Million Posts: A Data Set of German Online Discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1241–1244, Tokyo, Japan.

Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. Gottbert: a pure german language model. *Preprint*, arXiv:2012.02110.

Moritz Scherrmann. 2023. German finbert: A german pre-trained language model. *Preprint*, arXiv:2311.08793.

B Schölkopf, R Williamson, AJ Smola, and J Shawe-Taylor. 1999. Single-class support vector machines. In *Dagstuhl-Seminar 99121: Unsupervised Learning*, pages 19–20. Schloss Dagstuhl, Leibniz-Zentrum für Informatik.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. ISSN: 2379-190X.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. *arXiv:1508.07909 [cs]*. ArXiv: 1508.07909.

Daniel Solling. 2009. Små bokstäver ökade avståndet till tyskarna. Library Catalog: spraktidningen.se.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, USA. Association for Computational Linguistics. Event-place: Edmonton, Canada.

Cagri Toraman, Eyup Halit Yilmaz, Şahinuç Furkan, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv:1912.07076 [cs]*. ArXiv: 1912.07076.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

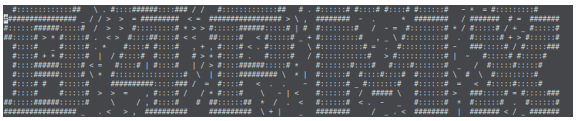
Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation. *Preprint*, arXiv:2109.04588.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. *arXiv:1904.00962 [cs, stat]*. ArXiv: 1904.00962 version: 5.

A Filtering OSCAR

Inside OSCAR interesting artefacts were found. Besides encoding errors, mainly affecting Umlauts, there were occurrences of ASCII arts (see Figures 1) and source code.



(a) B4PMX written in ASCII.

(b) Freemasonry sign.

(c) A turtle.

Figure 1: ASCII arts found in OSCAR.

B Model Properties

The number of parameters in BERT-like models varies significantly based on their architecture (see 4). The base version of BERT has approximately 110 million parameters, while the large version has about 340 million. RoBERTa, an optimized version of BERT, has 125 million parameters in its base model and 355 million in the large model, benefiting from extended training and larger datasets. The multilingual XLM-RoBERTa comes in two main versions: the base model with around 270 million parameters and the large model with about 550 million, which helps handle multiple languages effectively. Electra, using a generator-discriminator framework, achieves high performance with fewer

parameters, with the base model having about 110 million parameters and the large model around 335 million.

Model	Vocab Size	#Params
XLM-R$_{\text{large}}$	250002	559890432
$^{\text{f}}$GottBERT$_{large}$	52009	357145600
GBERT$_{\text{large}}$	31102	335735808
GELECTRA$_{\text{large}}$	31102	334686208
XLM-R$_{\text{base}}$	250002	278043648
mBERT	119547	177853440
$^{\text{f}}$GottBERT$_{base}$	52009	125985024
GBERT$_{\text{base}}$	31102	109927680
dbmdzBERT	31102	109927680
GELECTRA$_{\text{base}}$	31102	109337088
GermanBERT	30000	109081344

Table 4: The size of the vocabulary and the size of the parameters are shown for the model types used in this study. This table does not show other design differences of the models. Values were extracted using Huggingface's transformers library.

C Perplexity

During the model pre-training the perplexity of the model is computed based on a test set for each optimization cycle and based on a validation set at each checkpoint (see Figure 2). Within the training all the curves show a plateau: the base models only a short one, while the large models have a relatively long one. Some models even have upward spikes, which could be interpreted locally as divergence when observing the training process if they are not known. Furthermore, we see a very flat convergence of the models after 40k steps at the latest. This convergence can also be seen within the validation set based perplexity which was computed at each epoch.

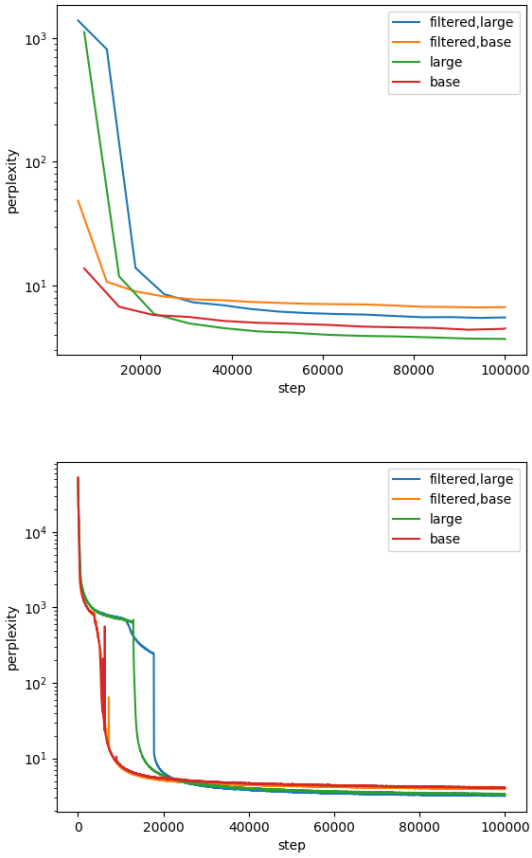


Figure 2: Perplexity of the GottBERT models. Top based on a validation at the checkpoints. Bottom based on the validation of each optimization cycle during the training.

D Parameters

The parameter space for our grid search is listed in Table 5. In addition, Table 6 shows the parameters of the best models (selection based on validation set) of the respective tasks. The time required for the evaluation is shown in Table 3. The tasks were computed on Nvidia Titan RTX and Nvidia A40 graphics devices and we relied on Huggingface's transformers library in version v4.34.1.

Parameter	Values
Learning Rate	5e-5, 2e-5, 1e-5, 7e-6, 5e-6, 1e-6
Batch Size	16, 32, 48, 64
Epochs	30

Table 5: Hyperparameters used in the grid search of the downstream tasks.

Model	GermEval 2014		CoNLL 03		GermEval 2018 coarse		GermEval 2018 fine		10kGNAD	
	BF	LR	BF	LR	BF	LR	BF	LR	BF	LR
$\mathrm{GottBERT}_{base}$	16	1 E-05	32	2 E-05	48	7 E-06	32	5 E-06	32	5 E-06
$\mathrm{GottBERT}^{\dagger}_{base}$	48	2 E-05	32	5 E-05	48	1 E-05	64	7 E-06	32	5 E-06
$^{\mathrm{f}}\mathrm{GottBERT}_{base}$	16	7 E-06	16	1 E-05	16	1 E-05	48	2 E-05	16	5 E-06
$^{\mathrm{f}}\mathrm{GottBERT}^{\dagger}_{base}$	16	1 E-05	64	5 E-05	16	1 E-05	16	2 E-05	16	1 E-05
$\mathrm{GELECTRA}_{\mathrm{base}}$	32	5 E-05	64	5 E-05	16	2 E-05	48	5 E-05	48	5 E-05
$\mathrm{GBERT}_{\mathrm{base}}$	16	2 E-05	64	2 E-05	32	1 E-05	16	5 E-05	16	2 E-05
dbmdzBERT	48	2 E-05	48	5 E-05	16	5 E-06	64	2 E-05	16	2 E-05
GermanBERT	32	2 E-05	16	1 E-05	16	1 E-05	32	1 E-05	32	5 E-05
$\mathrm{XLM\text{-}R}_{\mathrm{base}}$	64	2 E-05	16	1 E-05	48	5 E-05	64	5 E-05	48	2 E-05
mBERT	48	1 E-05	16	2 E-05	16	2 E-05	64	5 E-05	64	2 E-05
$\mathrm{GottBERT}_{large}$	64	5 E-06	16	5 E-06	64	5 E-06	32	7 E-06	64	1 E-06
$^{\mathrm{f}}\mathrm{GottBERT}_{large}$	32	5 E-06	48	2 E-05	32	5 E-06	32	7 E-06	16	5 E-06
$^{\mathrm{f}}\mathrm{GottBERT}^{\dagger}_{large}$	16	5 E-06	48	1 E-05	48	1 E-05	32	5 E-06	64	2 E-05
$\mathrm{GELECTRA}_{\mathrm{large}}$	16	7 E-06	16	5 E-06	64	1 E-05	32	2 E-05	32	2 E-05
$\mathrm{GBERT}_{\mathrm{large}}$	16	7 E-06	32	5 E-06	16	2 E-05	64	2 E-05	64	5 E-05
$\mathrm{XLM\text{-}R}_{\mathrm{large}}$	16	7 E-06	48	1 E-05	32	1 E-05	32	1 E-05	16	5 E-06

Table 6: Hyperparameters of the best downstream task model of the respective tasks and pre-trained models. BF is the batch size and LR the learning rate.

Task		Computation Time
XNLI		672:59
GermEval 2014		284:20
CoNLL03		169:26
GermEval 2018	coarse	113:36
	fine	113:47
10kGNAD		195:21

Table 7: Computation time in hours and minutes for the downstream tasks summing up to 1549 hours and 29 minutes which are approximately 64.6 days.