German's Next Language Model

Branden Chan[*][†], **Stefan Schweter**[*][‡], **Timo Möller**[†]

[†]deepset
{branden.chan, timo.moeller}@deepset.ai

[‡]Bayerische Staatsbibliothek München
Digital Library/Munich Digitization Center
stefan.schweter@bsb-muenchen.de

Abstract

In this work we present the experiments which lead to the creation of our BERT and ELECTRA based German language models, GBERT and GELECTRA. By varying the input training data, model size, and the presence of Whole Word Masking (WWM) we were able to attain SoTA performance across a set of document classification and named entity recognition (NER) tasks for both models of base and large size. We adopt an evaluation driven approach in training these models and our results indicate that both adding more data and utilizing WWM improve model performance. By benchmarking against existing German models, we show that these models are the best German models to date. Our trained models will be made publicly available to the research community.

1 Introduction

Deep transformer based language models have shown state-of-the-art results for various Natural Language Processing tasks like text classification, NER and question answering (Devlin et al., 2019). They are pretrained, first by feeding in large unlabeled text corpora before being fine-tuned on the downstream task. In this work we present a set of German BERT and ELECTRA models, the best of which, GELECTRA$_{Large}$, significantly improves upon state of the art performance on the GermEval18 hate speech detection task by about +4% / +2.5% for the coarse and fine variants of the task respectively. This model also reaches SoTA on the GermEval14 NER task, outperforming the previous best by over +4%. While performant, such models are prohibitively large for many and so we also present a new GBERT model which matches deepset BERT, the previous best German BERT, in size but outperforms it by +2.23% F1 averaged over three tasks.

In the process of pretraining the language models, we also a) quantify the effect of increasing the training data by an order of magnitude and b) verify that whole word masking has a positive effect on BERT models.

Because of the computational expense of training large language models from scratch, we adopt a downstream-oriented evaluation approach to ensure that we get the best performance from a limited number of runs. This involves regularly checkpointing the model over the course of pretraining, evaluating these on a set of classification and NER tasks and selecting as final the checkpoint which shows the best performance. This stands in contrast to approaches where the final model is simply saved after a fixed number of steps. Our method is also an important tool in diagnosing pretraining and we hope that it will be of use to other teams looking to train effective language models on a budget.

2 Related work

Modern language model architectures are trained to build word representations that take into consideration the context around a given word. First versions such as ELMo (Peters et al., 2018), ULMFiT

---

[*]Equal contribution.

(Howard and Ruder, 2018) and FLAIR (Akbik et al., 2018) are LSTM based and these were able to set new performance benchmarks on downstream tasks like text classification, PoS tagging and NER. More recent approaches use Transformer-based (Vaswani et al., 2017) architectures and examples include GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020) and ELECTRA (Clark et al., 2020).

In this work we focus on BERT and ELECTRA models. BERT uses a masked language modeling (MLM) strategy to corrupt an input sentence by replacing some tokens with a `[MASK]` symbol. The model is then trained to re-construct the original token. However, this method of training is somewhat restricted in that the model only learns from the masked out tokens which typically make up about 15% of the input tokens.

ELECTRA addresses this problem by introducing a new pretraining task called Replaced Token detection. Instead of masking out tokens, a subset of the input tokens are substituted by a synthetically generated token. The model is then trained to classify whether each input token is original or substituted, thus allowing for gradient updates at every input position. Practically speaking, this is achieved by having a discriminator that performs the replaced token detection and a generator which provides plausible token substitutes. These two components are trained jointly and are both Transformer based.

The BERT model received an update when the original authors added Whole Word Masking[1] whereby masking one subword token requires that all other tokens in the word are also masked out. The authors report that this method improves the training signal by removing the easiest cases and show that it improves performance in their tasks.

There is also a line of work that looks into bringing language modeling techniques that were first developed on English to other languages. These include but are not limited to monolingual models such as CamemBERT (Martin et al., 2020) and FlauBERT (Le et al., 2020) for French, Finnish BERT (Virtanen et al., 2019) and German BERTs by DBMDZ[2] and deepset[3]. For a more comprehensive list, see (Nozza et al., 2020).

Some models are also capable of supporting multiple languages such as multilingual BERT ($mBERT_{Base}$) and XLM-RoBERTa (Conneau et al., 2019). Multilingual BERT is a multilingual model for 104 different languages[4] trained on Wikipedia dumps. The XLM-RoBERTa model is trained on 2.5TB of data from a cleaned Common Crawl corpus (Wenzek et al., 2020) for 100 different languages.

It is worth emphasizing here that systems trained on naturally occurring data will learn pre-existing cultural biases around gender (Bolukbasi et al., 2016), race and religion (Speer, 2017). Critical evaluation of machine learning methods is more important than ever as NLP is gaining broader adoption. Researchers have been advocating for better documentation of decisions made during the construction of a dataset (Gebru et al., 2018), explicit statements of a dataset's "ingredients" (Holland et al., 2018) and recognition of the dataset characteristics that may lead to exclusion, overgeneralisation and underexposure (Bender and Friedman, 2018). These topics will be addressed in Section 3.1.

3 Datasets

3.1 Pretraining Data

We have available to us, a range of different German language corpora that we use in different combinations for our model pretraining. OSCAR (Ortiz Suárez et al., 2019) is a set of monolingual corpora extracted from Common Crawl. The Common Crawl texts are pre-processed (e.g. HTML entities are removed) and a language classification model is used to sort texts by language. We use the unshuffled version of the German OSCAR corpus, resulting in 145GB of text. The Wikipedia dump for German is preprocessed with the WikiExtractor[5] script forming a corpus of size 6GB. The OPUS project[6] (Tiedemann, 2012) has collected texts from various domains such as movie subtitles, parliament speeches and

---

[1] https://github.com/google-research/bert/commit/0fce551
[2] https://github.com/dbmdz/berts
[3] https://deepset.ai/german-bert
[4] https://github.com/google-research/bert/blob/f39e88/multilingual.md
[5] https://github.com/attardi/wikiextractor
[6] http://opus.nlpl.eu

Dataset	Size
OSCAR	145
OPUS	10
Wikipedia	6
OpenLegalData	2.4

Table 1: The size of each dataset in gigabytes.

books and these comprise a collection of around 10GB. From Open Legal Data[7] (Ostendorff et al., 2020) there is a dataset of about 2.4GB of German court decisions. Table 1 shows an overview over all datasets.

As discussed in Section 2, our pretrained language models will learn pre-existing biases from the training datasets. The main portion (89%) of our training data, namely the OSCAR dataset, uses texts scraped from the internet, which is in some respects problematic. First off, this dataset contains a lot of explicit and indecent material. While we filtered out many of these documents through keyword matching, we cannot guarantee that this method was successful in every case. Furthermore, many websites contain unverified information and any dataset containing this kind of text can lead to a skewed model that reflects commonly found lies and misconceptions. This includes gender, racial and religious biases which are found in textual data of all registers and so we advise that anyone using our model to recognise that it will not always build true and accurate representation of real world concepts. We implore users of the model to seriously consider these issues before deploying it in a production setting, especially in situations where impartiality matter, such as journalism, and institutional decision making like job applications or insurance assessments.

3.2 Downstream Data

3.2.1 GermEval18

For text classification we use GermEval18 (Coarse) and GermEval18 (Fine) which are both hate speech classification tasks (Wiegand et al., 2018). GermEval18 (Coarse) requires a system to classify a tweet into one of two classes: `OFFENSE` if the tweet contains some form of offensive language, and `OTHER` if it does not. GermEval18 (Fine) extends the coarse-grained task and contains four classes: `OTHER` for non-offensive tweets as well as `PROFANITY`, `INSULT` and `ABUSE` which are all subclasses of `OFFENSE` from the coarse variant of the task.

3.2.2 GermEval14

For NER, we use the GermEval14 (Benikova et al., 2014) shared task. The data is sampled from German Wikipedia and News Corpora and contains over 31,000 sentences and 590,000 tokens. The dataset is one of the largest NER datasets for German and features an advanced annotation schema that allows for nested annotations. The four main classes (`PERSON`, `ORGANISATION`, `LOCATION` and `OTHER`) each have part and derivative variants (e.g. `LOCpart` or `PERderiv`) resulting in 12 classes in total.

4 Training

4.1 Method

To train our German BERT and ELECTRA we use the Tensorflow training scripts from the official repositories[8]. We train models that match the size of the original BERT$_{Base}$, BERT$_{Large}$, ELECTRA$_{Base}$ and ELECTRA$_{Large}$. The hyperparameters used for training can be found in Table 2. The base models were trained on single Google Cloud TPUs v3 (8 cores) while large models were trained on pods of 16 TPUs v3 (128 cores).

---

[7]`http://openlegaldata.io/research/2019/02/19/court-decision-dataset.html`
[8]`https://github.com/google-research/bert` and `https://github.com/google-research/electra`

	GBERT$_{Base}$	GBERT$_{Large}$	GELECTRA$_{Base}$	GELECTRA$_{Large}$
max sequence length	512	512	512	512
batch size	128	2048	256	1024
warmup steps (k)	10	10	10	30
learning rate	1e-04	1e-04	2e-04	2e-4
checkpoint every (k)	100	100	76.6	100
max train steps (k)	4000	1000	766	1000
layers	12	24	12	24
hidden states	768	1024	768	1024
attention heads	12	16	12	16
vocab size (k)	31	31	31	31
train time (days)	7	11	8	7

Table 2: Hyperparameters for language model pretraining.

4.2 Models

In total, we trained 7 separate models with different combinations of data and model size as well as Whole Word Masking (WWM) for BERT models. The German DBMDZ BERT$_{Base}$, is the same size as BERT$_{Base}$ and was trained using the OPUS and Wikipedia corpora. It serves as our baseline model. We train four BERT variants of it, each referred to as GBERT, each using the same cased vocabulary as DBMDZ BERT$_{Base}$. These match BERT$_{Base}$ in size unless they have the "Large" suffix, in which case they match BERT$_{Large}$:

- GBERT$_{Data}$ - trained on all available data without Whole Word Masking

- GBERT$_{WWM}$ - trained on the same data as DBMDZ BERT$_{Base}$ but uses Whole Word Masking

- GBERT$_{Data + WWM}$ - trained on all available data and uses Whole Word Masking

- GBERT$_{Large}$ - trained on all available data and uses Whole Word Masking

We also trained three ELECTRA variants of DBMDZ BERT$_{Base}$, each referred to as GELECTRA models, which also match the size of the original ELECTRA$_{Base}$ unless they have the "Large" suffix in which case they match ELECTRA$_{Large}$:

- GELECTRA - trained on same data as DBMDZ$_{Base}$ BERT

- GELECTRA$_{Data}$ - trained on all available data

- GELECTRA$_{Large}$ - trained on all available data

The best models of each architecture and size are uploaded to the Hugging Face model hub[9] as *deepset/gbert-base*, *deepset/gbert-large*, *deepset/gelectra-base* and *deepset/gelectra-large*.

5 Evaluation

In our approach, models are evaluated continuously during pretraining. Model checkpoints are saved at regular intervals and converted into PyTorch models using Hugging Face's Transformers library (Wolf et al., 2019). Using the FARM framework[10], we evaluate the performance of each checkpoint on GermEval18 (Coarse) and GermEval18 (Fine) which are both hate speech classification tasks (Wiegand et al., 2018). Using Hugging Face's Transformers we also evaluate on GermEval14 (Benikova et al., 2014) which is a NER task.

---

[9]https://huggingface.co/models
[10]https://github.com/deepset-ai/FARM

	GermEval18 (Coarse)	GermEval18 (Fine)	GermEval14
Type	Classification	Classification	NER
Train Samples	4509	4509	24002
Dev Samples	501	501	2200
Test Samples	3533	3533	5100
Classes	2	4	12
Max Epochs	5	5	3
Max Train Steps	705	705	4500
Evaluation Every	50 steps	50 steps	1500 steps
Learning Rate	5e-06	5e-06	5e-05
Batch Size	32	32	16
Max Seq Len	150	150	128
Metric	F1 (macro)	F1 (macro)	F1 (micro)

Table 3: Details of the downstream tasks and hyperparameters for model finetuning for all three tasks.

In BERT, the vector corresponding to the `[CLS]` token serves as a representation of the whole input sequence, while in ELECTRA, all word vectors are combined through a feed forward layer. In both cases, this input sequence representation is passed through a single layer Neural Network in order to perform prediction. In the NER task, each vector corresponding to the first token in a word is passed through a single layer Neural Network and the resulting prediction is applied to the whole word.

Each checkpoint is evaluated 3 times on each document classification task since we observed significant variance across different runs. Each of these runs is performed with early stopping and a different seed each time. For NER, the model is evaluated just once without early stopping. The reported performance is the average of the single best run for GermEval18 (Coarse), GermEval18 (Fine) and GermEval14. Table 3 summarizes the most important details and parameters of each task. For all experiments, we use an Nvidia V100 GPU to accelerate training. For each model, we choose the checkpoint that shows the best performance.

For comparison, we also run this evaluation pipeline on the two publicly available German BERT models (deepset German $BERT_{Base}$ and DBMDZ German $BERT_{Base}$) as well as multilingual models such as $mBERT_{Base}$ and $XLM-RoBERTa_{Large}$.

6 Results

The downstream performance graphs in Figure 1 show that the models are capable of learning with most of the gains being made in the first phase of training and more incremental gains coming later. The best checkpoints come at different points for different models as can be seen in Table 4.

In Table 5 are the evaluation results for each model's best checkpoint for each of the three downstream tasks with comparison to benchmark models and previous SoTA results. For GermEval18, results from the best-performing systems are reported (Wiegand et al., 2018). For GermEval14 we report the result that can be achieved using the FLAIR framework (Akbik et al., 2019).

	Steps (k)
$GBERT_{Data}$	3900
$GBERT_{WWM}$	1500
$GBERT_{Data + WWM}$	2000
$GBERT_{Large}$	900
GELECTRA	766
$GELECTRA_{Data}$	766
$GELECTRA_{Large}$	1000

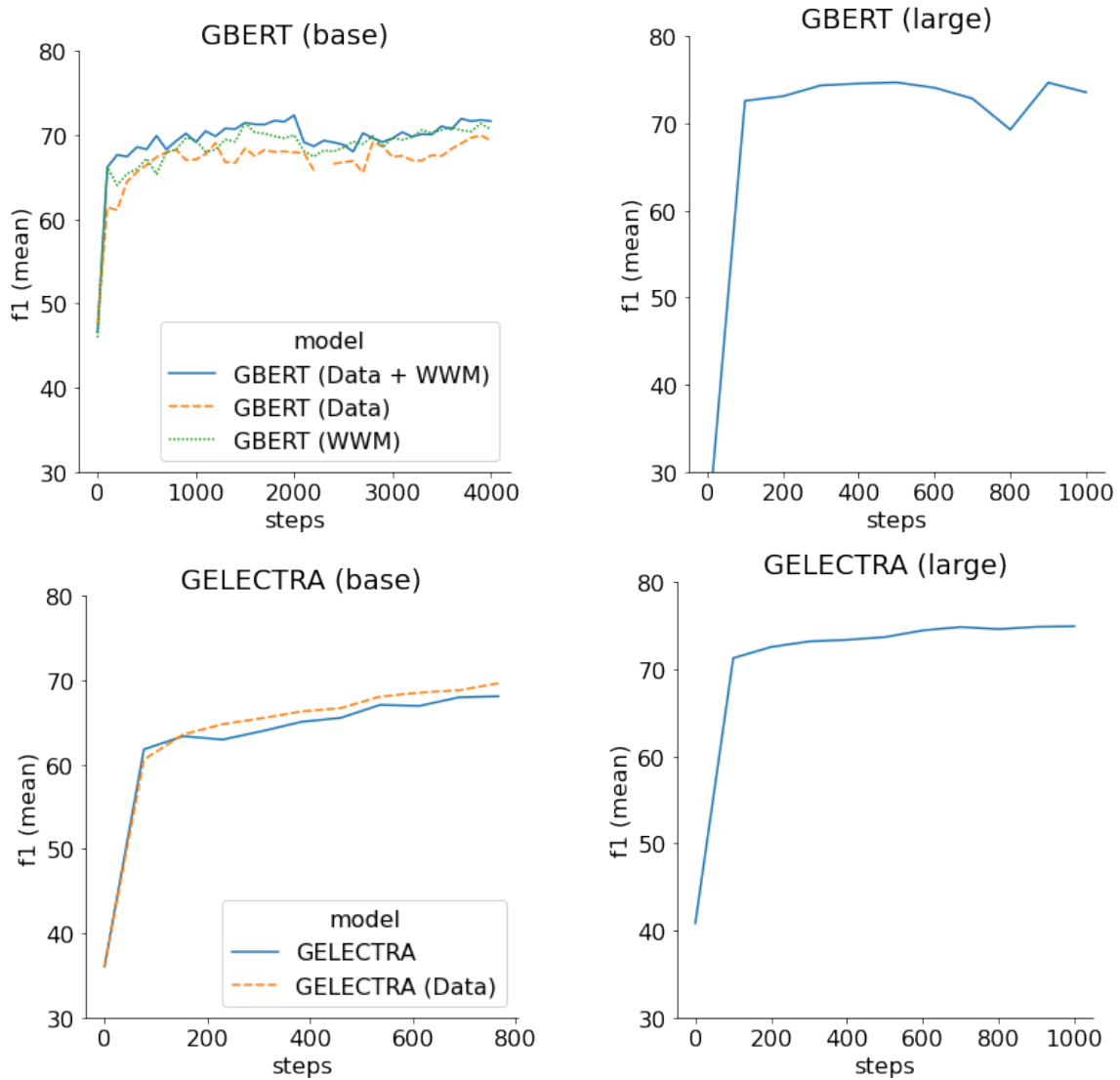Table 4: Best checkpoint of each trained model.

Figure 1: The F1 performance of each model averaged over the three downstream tasks over the course of language model pretraining.

In GermEval18 (Coarse), GBERT$_{\text{Data + WWM}}$, XLM-Roberta$_{\text{Large}}$, GBERT$_{\text{Large}}$ and GELECTRA$_{\text{Large}}$ all improve upon the previous SoTA. GELECTRA$_{\text{Large}}$ does so with the largest margin reaching a score that is +3.93% better. In GermEval18 (Fine), XLM-Roberta$_{\text{Large}}$ beats the previous best by +1.39% and GELECTRA$_{\text{Large}}$ sets a new SoTA that is better than the previous by +2.45%. In GermEval14, all 7 trained models exceed the previous SoTA, with GELECTRA$_{\text{Large}}$ showing a +4.3% improvement over the previous best.

These results indicate that adding extra data gives a consistent but modest performance boost to our language models. GBERT$_{\text{Data}}$ outperforms DBMDZ BERT$_{\text{Base}}$ by +0.25%, GBERT$_{\text{Data + WWM}}$ outperforms GBERT$_{\text{WWM}}$ by +0.93% and GELECTRA$_{\text{Data}}$ outperforms GELECTRA by +1.59%. For the BERT models, Whole Word Masking also shows a consistent positive impact with GBERT$_{\text{WWM}}$ outperforming DBMDZ BERT$_{\text{Base}}$ by +1.70% and GBERT$_{\text{Data + WWM}}$ outperforming GBERT$_{\text{Data}}$ by +2.38%.

7 Discussion

7.1 Model Size

The large models that we train show much stronger performance than the base models. GBERT$_{\text{Large}}$ outperforms GBERT$_{\text{Data + WWM}}$ by +2.33% averaged F1 and GELECTRA$_{\text{Large}}$ outperforms GELECTRA$_{\text{Data}}$

	Params	GermEval18 (Coarse)	GermEval18 (Fine)	GermEval14	Averaged F1
DBMDZ BERT$_{\text{Base}}$	110m	75.23	47.39	87.90	69.72
deepset BERT$_{\text{Base}}$	110m	74.7	48.8	86.87	70.12
mBERT$_{\text{Base}}$	172m	70.00	45.20	87.44	67.55
XLM-Roberta$_{\text{Large}}$	550m	78.38	54.1	87.07	73.18
GBERT$_{\text{Data}}$	110m	74.51	48.01	87.41	69.97
GBERT$_{\text{WWM}}$	110m	76.48	49.99	87.80	71.42
GBERT$_{\text{Data + WWM}}$	110m	78.17	50.90	87.98	72.35
GBERT$_{\text{Large}}$	335m	80.08	52.48	88.16	73.57
GELECTRA	110m	76.02	42.22	86.02	68.09
GELECTRA$_{\text{Data}}$	110m	76.59	46.28	86.02	69.63
GELECTRA$_{\text{Large}}$	335m	**80.70**	**55.16**	**88.95**	**74.94**
Previous SoTA		76.77 (TU Wien)	52.71 (uhhLT)	84.65 (Flair)	

Table 5: Downstream evaluation results for the best checkpoints of each GBERT and GELECTRA model compared to a set of benchmark models. For GermEval18 we report scores for the best-performing systems (Wiegand et al., 2018), and the result reported by Flair framework (Akbik et al., 2019) for GermEval14.

by +5.31%. It must be noted however, that their differing training regimes mean that the large models are trained on many more tokens than their base counterparts. In future, we would also be interested in training larger models with less data in order to better quantify the gains that come from model size and the gains that come from the extra data.

7.2 Training Length

From the downstream evaluation graphs in Figure 1, it is clear that the models gain most of their performance after a relatively short amount of training steps. GBERT$_{\text{WWM}}$ and GBERT$_{\text{Data + WWM}}$ both show an upward trend in the second half of model training suggesting they could still benefit from continuing training. There is also a clear upward trend over the course of GELECTRA and GELECTRA$_{\text{Data}}$'s training suggesting these models are undertrained. It should also be noted that none of the models exhibit any clear signs of overfitting or performance degradation and may improve with further training.

7.3 ELECTRA Efficiency

One of the central claims of the ELECTRA paper is that it is capable of learning more efficiently than MLM based Language Models. This is exemplified by the comparison of GBERT$_{\text{Large}}$ and GELECTRA$_{\text{Large}}$. By the end of their 1 million steps of training, GELECTRA$_{\text{Large}}$ has only seen half the number of tokens that GBERT$_{\text{Large}}$ due to its smaller batch size and yet outperforms it by +1.47% averaged F1.

7.4 Instabilities

The dip in performance around 2 million steps for the base sized GBERT models (See Figure 1) happens to coincide with our training regime whereby the model training is stopped, saved and then reloaded at 2 million steps. While we suspect that these two events are related, it was beyond the scope of this project to investigate the exact reasons.

8 Conclusion

The set of German models which we trained vary in terms of training regime and model architecture. We hope that the results that we present here will serve as important data points to other NLP practitioners who are looking to train language models from scratch but are limited by compute. Our experiments should give other teams a sense of the batch sizes and training lengths that make for efficient model training. On top of this, we also present a set of GELECTRA and GBERT models which, according to our evaluations, set new SoTA performance for both large and base sized models on GermEval18 and GermEval14.

Acknowledgements

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. 2014. Germeval 2014 named entity recognition: Companion paper. *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition, Hildesheim, Germany*, pages 104–112.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4349–4357. Curran Associates, Inc.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

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, June. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, III Daumé, Hal, and Kate Crawford. 2018. Datasheets for Datasets. *arXiv e-prints*, page arXiv:1803.09010, March.

Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. The Dataset Nutrition Label: A Framework To Drive Higher Data Quality Standards. *arXiv e-prints*, page arXiv:1805.03677, May.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France, May. European Language Resources Association.

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 preprint arXiv:1907.11692*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte 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*.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? Making Sense of Language-Specific BERT Models. *arXiv e-prints*, page arXiv:2003.02912, March.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In Piotr Bański, Adrien Barbaresi, Hanno Biber, Evelyn Breiteneder, Simon Clematide, Marc Kupietz, Harald Lüngen, and Caroline Iliadi, editors, *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom, July. Leibniz-Institut für Deutsche Sprache.

Malte Ostendorff, Till Blume, and Saskia Ostendorff. 2020. Towards an open platform for legal information. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, page 385–388, New York, NY, USA. Association for Computing Machinery.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Robyn Speer. 2017. Conceptnet numberbatch 17.04: better, less-stereotyped word vectors.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz 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 e-prints*, page arXiv:1912.07076, December.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France, May. European Language Resources Association.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018*, pages 1 – 10. Austrian Academy of Sciences, Vienna, Austria.

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