Should You Mask 15% in Small Language Modelling?

Katharina Strauss Soegaard and **Aaron Chao** and
Max Huppertz and **Timo Stephan Hermann Stoll**

Abstract

After the success of BERT the choice of a 15% masking rate for the pre-training of Language Models has been widely adopted (Wettig et al., 2023). But this arbitrary choice of masking rate may not be the best choice for every model. With recent research into the area of masking rates (Wettig et al., 2023), results indicate that larger models can benefit from higher masking rates However, this research was only conducted with English data sets. Our approach focuses on establishing a masking rate guidance for small-scale models such as BERT Mini and conducting experiments using German language modeling tasks. We were able to show that similar to the results found by (Wettig et al., 2023) a higher masking rate does not diminish the prediction accuracy of small scale language models. Furthermore, there appears to be no significant difference in the effect of masking rates between English and German language tasks as they show similar behavior. Therefore, we were able to replicate the findings by adapting the masking rate to small language models. We also provide results on German benchmark data sets to evaluate the impact of the masking rate on model performance.

1 Introduction

Masked Language Modeling is a bidirectional modeling method that predicts a masked token in a sequence. However, the hyper parameter of masking rate has under examined, and has ubiquitously been set at 15% for most of BERT's successors, which is not ideal, especially for larger language models as found by (Wettig et al., 2023).

In this research, we hope to examine whether these results can be extrapolated to smaller language models, and examine how different masking rates perform on smaller language models as compared to larger models.

Smaller models were utilized for two purposes. Firstly, it was assumed that smaller models would be impacted at a different degree than larger models, and hence, the ideal masking rate and impact of different masking rates would also be significantly different than those for larger models. Secondly, the training time for smaller models is significantly lower than comparable larger models, which allowed for faster acquisition of results for analysis.

Furthermore, German was chosen as the language of the data sets to examine whether these results would be applicable to another language. German, as compared to English, also has a more complicated syntactical structure due to its non-projective syntax, where its syntactical structure is not draw-able without crossing two arcs, and different grammar and lengthier words. In addition, there is less research into German LLMs than towards English LLMs and their properties.

2 Approach

In order to faciliate our approach of training a Bert Mini model on German language data we used a multiple step approach to pretraining and evaluating on downstream tasks.

2.1 Replicating the RoBERTa baseline model

First of all, we replicated the results of the RoBERTa baseline model given by (Liu et al., 2019) using the originally given hyperparameter. Since the given scripts did not work for us we replicated them using the huggingface transformer library (Wolf et al., 2020) as well as fairseq (Ott et al., 2019). The pretrained model was then evaluated on the GLUE benchmark dataset (Wang et al., 2018). The results can be found in table 1. Since the RoBERTa baseline model proved to be infeasible for pretraining due to its large number of parameters (around 125 million) (Liu et al., 2019) we resorted to using a small scale model. Our model of choice is given by the BERT Mini model introduced by (Turc et al., 2019) who used it in a teacher student setting. Since BERT Mini only has four instead of twelve transformer layers as well as a smaller number of attention heads and neurons in the hidden layers the number of parameters reduce to about 10% of the baseline model (Turc et al., 2019). This enables us to pretrain our own model using different masking rates while retaining a feasible computation time for our project. Since the effect of the masking rate was not explored for small scale models in the baseline paper (Wettig et al., 2023) we aim to give insights into the area of small scale model research. We further fine-tuned a DistilBERT (Sanh et al., 2020) model on our German benchmark datasets to have a comparable baseline for our small scale model since the original paper only provides results for English datasets.

2.2 Pretraining a BERT Mini model on German language data

With our choice of the Wikipedia dataset for pretraining as outlined in section 3.1 the dataset was first preprocessed using a pretrained tokenizer. We tested multiple available tokenizers thorugh huggingface but did not observe a significant difference in their speed or performance. Therefore, we chose a tokenizer provided by the Bavarian State Library on huggingface which was ready to implement in our own pretraining pipeline. In total we pretrained four different models with differing masking rates and each pretraining took around 24 hours of computation time with the provided resources. Concerning the choice of hyperparameters we went with the choices outlined in the BERT Mini paper (Turc et al., 2019) as they provided us good results in the pretraining and trained the models for three epochs each as at that point we evaluated no further significant change in the training evaluation (fig 4.2). After pretraining we evaluated the training statistics received during the run which can be seen in table 2. These pretrained models were then used in the evaluation highlighted in section 4.

3 Data & Experiments

As our experiments require the usage of German language datasets we evaulated several datasets for our purposes. All of the following experiments are based on these datasets outlined in the following.

3.1 Choice of datasets

Before beginning the actual pretraining and evaluation of our BERT Mini models we first had to choose the datasets that we plan on using for both tasks. For the pretraining on masked language modeling we used the Wikipedia dataset with the German corpus version of 2022 since that dataset has been used in other German LLM approaches and proved to be successful for pretraining (Chan et al., 2020). This dataset consists of roughly 16GB of unlabelled training data seperated into the available German Wikipedia articles. Since the topics of Wikipedia are quite varied we believed this dataset to be a useful baseline for training. Originally we also planned on using the OSCAR dataset (Ortiz Suárez et al., 2020) and its German subcorpus but with the time and computational power constraints in the project it proved infeasible to use it for training data since the German corpus has a size of 145GB which proved too large for efficient pretraining. In addition, we chose three different datasets for the downstream task evaluation. All of these datasets are common benchmark datasets in German NLP tasks and have been used for evaluation of a variety of German LLMs such as GBERT (Chan et al., 2020) and others. In the area of text classification tasks we used two datasets. GermEval18 (Risch et al., 2018) and Gnad10k (Schabus et al., 2017). Both are classification tasks with a multi-class task and GermEval18 also includes a binary classification task. In the case of Gnad10k the task is to predict the category of German news articles sourced from online newspapers in one of ten different categories. In contrast to this GermEval18 is a classification task to predict offensive content in German tweets. The binary (coarse) classification task simply differentiates between offensive and non-offensive content whereas the fine grained task considers different types of offensive content such as insults or abusive content. Apart from classification our benchmark datasets also included GermEval14 (Darina Benikova, 2014), a dataset based on the task of Named Entity recognition (NER) which is also based on German news articles. Here the goal is to determine which part of a given sentence is the Named Entity and which parts do not constitute to this entity. All of these datasets were preprocessed by using the huggingface preprocessing pipeline and tokenized using the same tokenizer as for model pretraining.

4 Results

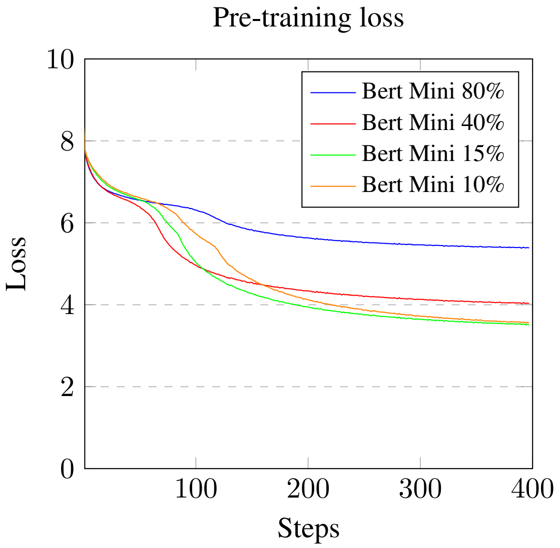
In the following section we will present the results of both the replication of Wettig et al. as well as the results of our own research using German Bert Mini models.

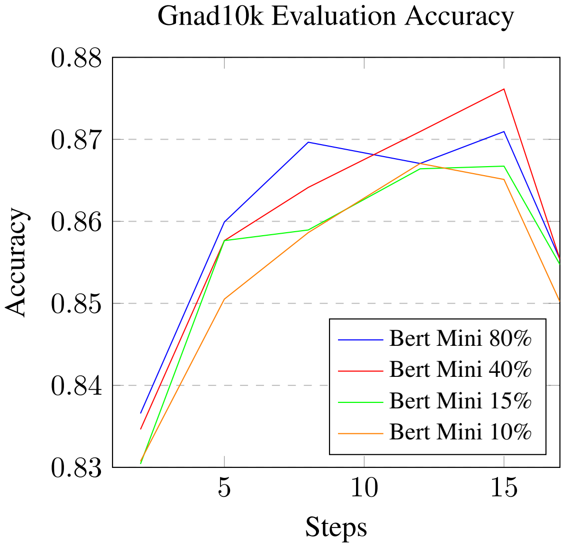
4.1 Replication

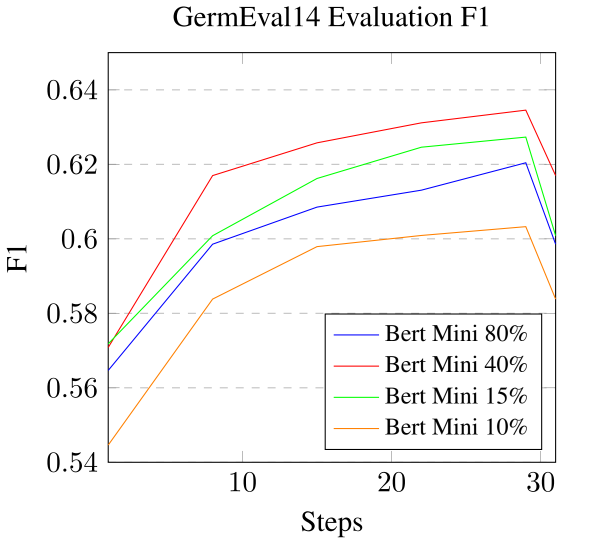
As we already mentioned we reproduced the results by Wetting et al. with their given pretrained models and finetuning them on the GLUE benchmark data set. Our results, the bold values in table 1, are very similar to the results from the paper (Wettig et al., 2023). Only in MRPC we achieve higher results and in RTE slightly lower results. Among all of the downstream tasks the RoBERTa baseline model pretrained with a masking rate of 40% showed the best performance. Therefore, the results from our replication prove the thesis that a higher masking rate (especially 40%) can improve the model performance on downstream tasks and 15% masking rate is indeed not the optimal masking rate for large language models.

4.2 German Models

We pre-trained four Bert Mini models using the masking rates 10%, 15%, 40% and 80% for each three epochs. In graph 4.2 the pre-training loss values can be seen. Two aspects are interesting from this graph. First of all, the model trained with a masking rate of 80% has a significantly higher loss value throughout and at the end of the pre-training. The other models have quite similar loss values with the Bert Mini model trained with a masking rate of 40% having a slightly higher value than the models with 10% and 15%. Those values however are understandable considering the fact that the model with 80% masking rate has a higher chance of wrongly predicting tokens when 80% of the tokens are masked which leads to a higher loss value. Secondly, the Bert Mini model with 40% masking rate converges faster than the other models which indicates a more efficient training than the other models. The following graph shows the average evaluation accuracy for each model finetuned on the Gnad10k data set with three random seeds. We can observe that the all of the models expect for the model with a masking rate of 10% achieve a very similar accuracy after 3 epochs. However the model with a masking rate of 80% has the highest performance in the first half of the training and the model with a masking rate of 40% in the second half of the finetuning. In our experiments with the GermEval14 data set we can observe slightly different behaviour. The Bert Mini model with a masking rate of 40% achieves the highest F1 score throughout the hole finetuning process. The model with a masking rate of 10% performs significantly worse and the models with 80% and 15% masking rates results in a very similar result.







In experiments with the GermEval18 data set

Table 1: Replication results (Wettig et al., 2023)

Masking rate	QNLI	RTE	SST2	MRPC	CoLA	STS_B
15%	**90.8** / 90.9	**64.3** / 67.3	**92.4** / 93.3	**85.5** / 77.0	**57.8** / 59.2	**86.7** / 87.7
40%	**91.2** / 91.6	**62.1** / 67.0	**92.3** / 92.8	**87.5** / 76.9	**59.3** / 61.0	**86.7** / 88.2
80%	**87.8** / 87.9	**56.7** / 58.6	**89.9** / 90.5	**81.1** / 72.1	**37.7** / 38.7	**85.5** / 86.3

using the fine labels we observed that the model with a masking rate of 15% has a slightly higher F1 score than the model with 80% masking rate. The model with 10% masking rate again performs the worst and the model with a masking rate of 40% has the highest initial score but at the end performs very similar to the model with 10%.

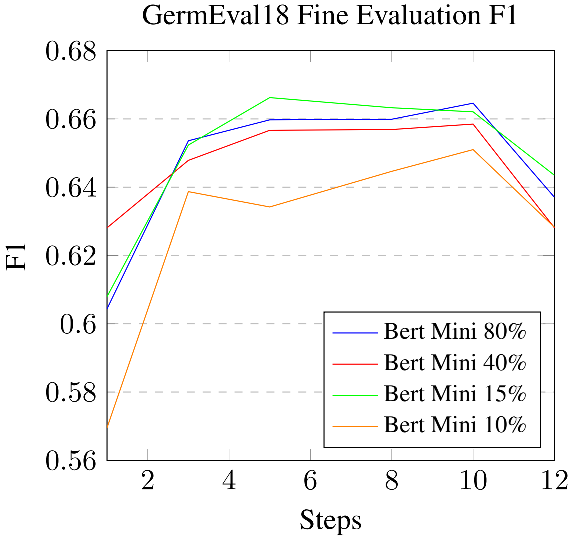


Table 2 shows an overview over all results of our tests. We performed the finetuning for each of the models three times with random seeds and calculated the average for each score. We used the F1 and precision score as metrics for the GermEval18 Coarse and Fine experiments because we noticed that the amount of the occurrences of the different classes varied drastically and we therefore wanted to analyze the precision of the models. In the finetuning on GermEval14 we used the F1 and recall score and for the Gnad10k finetuning we used accuracy as a metric. The best score for each dataset is highlighted in bold. We also fintuned an already pretrained German version of DistilBERT using the same finetuning process to use it as a comparison. The DistilBERT model has around 66 million parameters compared to the Bert Mini models which have 11.4 million parameters each.

Our results show that among all data sets the Bert Mini model with a masking rate of 40% performed the best. We could also observe that the model trained with a masking rate of 80% still performs very well compared to the lower models with lower masking rates and the model pre-trained with a masking rate performed the worst among all data sets. The DistilBERT model achieves a slightly higher performance when finetuned on the Gnad10k data set and singificantly higher performance on the GermEval14 and GermEval18 Fine data sets. However when finetuned on the GermEval18 Coarse data set all of our Bert Mini models achieve a higher F1 and precision score.

5 Discussion

Based on the results we found that higher masking rates do not reduce the performance significantly, and that having a masking rate of 40% performed the best across the different downstream tasks. Our model performed worse than the baseline in 2/4 tasks, but outperformed the baseline in the coarse GermEval18 task. Our model performed similar to the baseline for the GNAD10k. The baseline DistilBERT model has five times the parameters of our model, but despite the size difference our model showed good results in the different tasks. The two tasks our model struggled with were the GermEval18 and GermEval14. The GermEval18 task has four classes, which are split unevenly, and the GermEval14 task has 12 classes. The larger number of classes and uneven splits might be part of the reason why our model performs worse as it is too small. But despite having worse results than the baseline it still shows the same trends for the masking rate, and that having the higher masking rates improves the performance. These results are in contrast to the paper, (Wettig et al., 2023), which was only able to show these results for large models. Despite changing the language to German the model still behaved achieved higher accuracies when increasing the masking rate, despite the more complex structure and word length in German. Therefore this method should be explored in even more languages, also ones that are not from Germanic descent.

Table 2: Results Finetuning

Model	**GermEval18 Coarse**	**GermEval18 Fine**	**GermEval14**	**Gnad10k**
Bert Mini 10%	0.7184 / **0.7205**	0.651 / 0.6404	0.6033 / 0.6579	0.8651
Bert Mini 15%	0.7206 / 0.7193	0.6621 / 0.6523	0.6273 / 0.6776	0.8667
Bert Mini 40%	**0.7223** / 0.7204	0.6585 / 0.6486	**0.6346 / 0.6845**	**0.8762**
Bert Mini 80%	0.719 / 0.7184	**0.6646 / 0.6537**	0.6204 / 0.6735	0.87082
DistilBERT	0.7054 / 0.7055	0.7652 / 0.7668	0.8270 / 0.8423	0.8901

Some of the limitations that we faced throughout this project, were that we only were able to train the smaller models due to time constraints. While we did get good results, larger models for German was left unexplored. It would have been interesting to compare the accuracies across model sizes, to see if the masking rate had the same influence for all model sizes. We were only able to train on one data set. We wished to be able to use additional and especially bigger data sets, like OSCAR, which could have been done with more time and also for larger models. We also faced some issues with not having that many downstream task data sets in German, that had benchmark results in literature as well.

From our experiments we concluded that higher masking rates can help smaller models, and propose treating the masking rate as a hyperparameter could be beneficial and increase accuracy. The models with higher masking rates trained slightly faster and converges earlier compared to the lower masking rates. We were able to reproduce the results from the paper for smaller models, and in a more complex language. We had some limitations due to model size and available training data and downstream tasks, but were able to get comparable results. Since the higher masking rate did not improve the models in all cases, it might be beneficial to treat as a hyperparameter instead of just setting it to 15%. It should be tailored to the specific task the model is meant to perform, or adjusted to ensure the model performs well for all tasks.

References

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.

Max Kisselew Sebastian Padó Darina Benikova, Chris Biemann. 2014. Germeval 2014 named entity recognition shared task: Companion paper.

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.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

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.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.