Using Selective Masking as a Bridge between Pre-training and Fine-tuning

Tanish Lad
IIIT Hyderabad
tanish.lad@research.iiit.ac.in

Himanshu Maheshwari
Adobe Research[*]
him.maheshwari1999@gmail.com

Shreyas Kottukkal
IIIT Hyderabad
shreyas.shankar@students.iiit.ac.in

Radhika Mamidi
IIIT Hyderabad
radhika.mamidi@iiit.ac.in

Abstract

Pre-training a language model and then fine-tuning it for downstream tasks has demonstrated state-of-the-art results for various NLP tasks. Pre-training is usually independent of the downstream task, and previous works have shown that this pre-training alone might not be sufficient to capture the task-specific nuances. We propose a way to tailor a pre-trained BERT model for the downstream task via task-specific masking before the standard supervised fine-tuning. For this, a word list is first collected specific to the task. For example, if the task is sentiment classification, we collect a small sample of words representing both positive and negative sentiments. Next, a word's importance for the task, called the word's task score, is measured using the word list. Each word is then assigned a probability of masking based on its task score. We experiment with different masking functions that assign the probability of masking based on the word's task score. The BERT model is further trained on MLM objective, where masking is done using the above strategy. Following this standard supervised fine-tuning is done for different downstream tasks. Results on these tasks show that the selective masking strategy outperforms random masking, indicating its effectiveness.

1 Introduction

Pre-trained language models have been instrumental in spurring many advancements across various NLP tasks [27, 2, 16]. Most of these works follow the paradigm of unsupervised pre-training on large general-domain corpora followed by fine-tuning for downstream tasks, following the success of BERT [8]. While the pre-training and fine-tuning paradigm has proven to be highly successful, Gururangan et al. [11] has shown that even large LMs struggle to capture the complexity of a single textual domain. Thus, they suggest domain-specific pre-training on task-relevant corpora for better downstream performance in that domain. Other works [2, 4, 13, 3, 18] show improvements by pre-training BERT like models on huge in-domain corpora relevant to the downstream tasks.

BERT uses the Cloze task [25] for their masked LM (MLM) pre-training procedure wherein 15% of input tokens are masked from each sequence at random. Clark et al. [5] and Gu et al. [10] offer alternatives to this random masking strategy. Clark et al. [5] argues that learning a random 15% of the input sequence is inefficient and suggests a "replaced token detection" pre-training task instead. Gu et al. [10] proposes a selective-masking strategy, arguing that there are task-specific tokens that are more important to mask than other tokens. They added a task-guided selective masking pre-training

[*] The author was at IIIT Hyderabad during the work.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

between general pre-training and fine-tuning to learn the domain-specific and task-specific language patterns. This strategy makes use of a small in-domain dataset for selective masking.

In this work, we propose a novel way to tailor the BERT model for the downstream task via task-specific masking before the standard supervised fine-tuning. We take a pre-trained BERT model and tailor it for downstream tasks using task-specific selective masking on a small chunk of BookCorpus (Zhu et al. [28]). This tailored model is then fine-tuned for downstream tasks. Our method includes a novel approach to find tokens important for the downstream task using a list of seed words relevant to the downstream task. The word list and word embeddings are used to compute a 'task' score for each word, which is used to calculate the masking probability of the word. We experiment with different masking functions and show considerable value in such selective masking. Unlike Clark et al. [5], we use selective masking to train our model. Compared to Gu et al. [10], we use only word-lists instead of in-domain data for selective masking. Since BERT also uses BookCorpus, we do not use any new corpus for training. Collecting a word list is an easier alternative when in-domain datasets are scarce. We also extend this approach to a wider variety of downstream tasks than the classification tasks explored by Gu et al. [10].

Experimental results on different downstream tasks such as **sentiment analysis, hate speech classification, formality detection and NER on informal text** show that this methodology performs better than random masking. Thus our methodology is both effective and generalizable.

2 Task Specific Masking

In this section, we first describe the method to calculate the task score of the word. Following that, we describe different masking functions that use this task score to assign a probability to each word. The word is then masked with this probability.

2.1 Calculating Task Specific Score of a Word

We leverage the classification framework in Niu et al. [19], who calculated the formality score of a word for machine translation. We begin with a set of seed words indicative of different classes in downstream tasks. For example, in sentiment analysis, we choose negative-sentiment and positive-sentiment words as two sets of word-lists. Here we assume that the classification is binary (although we experiment with multi-class sentiment classification too). For another example, let us consider a non-classification task like NER on the domain of informal texts. Here our first list represents in-domain vocabulary and contains words usually found in informal texts. Our second list contains words not found in that domain, and thus here, we use words found in the formal texts.

Following Niu et al. [19], a Support Vector Machine (SVM) model is trained by assigning scores of 0 and 10 for the two sets of words, and a separating hyperplane is learned between Word2Vec [17] vector space representations of the two classes of seed words. Niu et al. [19] reported the best performance using an SVM model with Word2Vec representations. We did some experiments with GloVe embeddings [22] and SVM as well but word2vec gave better results. Once the SVM model is trained, euclidean distance to this hyperplane is used to measure the given word's task-specific score.

2.2 Masking Functions

As described earlier, our approach uses a masking function to compute the probability of masking a given word based on its task score (obtained from the SVM model described earlier). If the downstream task is classification, then we mask both the extremes. For example, if the task is sentiment classification, the language model is required to understand both extremes, viz. positive sentiments and negative sentiments. So we assign a high probability to both extremes. If the task requires the model to learn only one extreme, say NER on the informal dataset, then in that case, we assign a high probability to only one extreme (i.e., the one representing informality). In the following functions, s is the euclidean distance of the word from the SVM hyperplane. α, β, and γ are constants that were adjusted such that approximately 15% of the tokens are masked.

Masking Function 1: Step Function

$$f(s) = \begin{cases} 1 & s \leq \alpha \; or \; s \geq 10 - \alpha \\ 0 & otherwise \end{cases}$$

Masking Function 2: Linear Function

$$f(s) = \frac{max\,(\,s\,,\,10 - s\,) - \alpha}{\beta}$$

Masking Function 3: Concave Up (exponential)

$$f(s) = \frac{e^{\alpha.max(s,10-s)} - e^{\beta}}{\gamma}$$

The step function only masks words at the extreme and completely discards other words. However, Liu [14] has argued that the appearance of an opinion word in a sentence does not necessarily mean that the sentence expresses a positive or negative opinion. Similarly, Pavlick and Tetreault [21] has argued that formality is not only expressed by the use of formal/informal words but also "neutral" words, punctuations, capitalization, paraphrasing, etc. So in the remaining masking functions, we assign a non-zero probability to non-extreme words. The linear function assigns each word a probability linearly proportional to its importance. The exponential function glorifies the values which are towards the extreme and exponentially decreases the values as we move towards the center. It assigns each word a probability exponentially proportional to its importance.

3 Experiments

We take a pre-trained BERT model released by Devlin et al. [8] and further train it using different masking functions discussed above. Like the original BERT framework, we mask the chosen word 80% of the time, replace it with a random word 10% of the time, and leave it unchanged 10% of the time. The original BERT model uses WordPiece tokenization [26] and masks only tokens. Later whole word masking was introduced, where all the tokens associated with a word are masked. Since our task demands whole word masking, and for a fair comparison, we compare our results with both variants of random masking (whole word masking (WWM) and token masking (TM)).

Dataset	**Train**	**Test**
Amazon Review	40,000	5,000
Movie Review	8,534	2,128
Gab	28,776	5,000
Reddit	17,314	5,000
GYAFC - Family	103,934	2,664
OC	8,730	1,400
Humour	100,000	10,000
BTC	5,551	4,000
WNUT-17	3,394	2,296

Table 1: Dataset Statistics

We use a small chunk ($\sim$ 100 Mb) of BookCorpus [28] and train the BERT model on it for 20k steps using the above masking strategy. For a fair comparison, we also pre-train another BERT on this corpus using random masking (both token masking and whole word masking). To compare our results with Gu et al. [10], we also train BERT on the Amazon Reviews dataset using our selective masking and finetune it on the Movie Reviews dataset (discussed below). We test our approach on a variety of downstream tasks. We finetune for ten epochs and report the best result.

Masking Function	Sentiment Analysis			H.S.D		Formality Analysis		H.D.	NER	
	Amazon	**MR**	**MR+A.**	**Gab**	**Reddit**	**OC**	**Family**	**ColBERT**	**BTC**	**WNUT-17**
Random (TM)	59.46	85.66	87.43	92.22	88.92	85.21	87.88	**98.68**	49.99	18.80
Random (WWM)	59.79	85.47	87.05	92.26	87.88	84.86	88.25	98.45	49.93	19.25
MF 1	60.20	85.66	87.52	92.22	88.44	85.43	87.95	98.57	48.33	20.71
MF 2	**61.42**	86.58	88.34	93.14	**89.99**	86.57	89.18	98.64	50.26	21.04
MF 3	60.53	**87.60**	**89.65**	**93.20**	89.34	**87.43**	**89.03**	98.61	**51.78**	**21.96**

Table 2: Results on different downstream tasks. We report F-1 score for the NER task. MR+A. means Movie Reviews with Selective Masking on Amazon Reviews data, H.S.D. means Hate Speech Detection, H.D. means Humor Detection, and M.F. means Masking Function.

For **sentiment classification** task we use Amazon review dataset [12] and movie review dataset (referred as MR) [20]. Amazon review is a multiclass classification problem, while the movie review dataset is a binary classification. For **hate speech detection** we use data from Gab and Reddit [23] in a binary classification setup. For **formality detection** task we use GYAFC dataset [24] and online communication (OC) data [21]. For **NER on informal text** we use text from social media as a proxy to informal text. To this end, we use the dataset provided by Broad Twitter Corpus (referred as BTC) [6] and WNUT-17 [7]. For **humor detection** task we use the dataset by Annamoradnejad [1]. To also test whether the gain in results is due to the actual selective masking strategy and not any other reason, we deliberately use a word list whose words mismatch with the type of words present in the dataset for the humor detection task. The words present in the word list were the words commonly used in off-color humor (dark sexist, racial, etc. jokes). In contrast, the fine-tuning dataset consisted of short formal text with no vulgarity. We expect that there will be no significant difference in results using selective masking as compared to random masking. Table 1 provide statistics about each of these dataset. Note that we report the combined test and validation dataset as our test dataset.

For the sentiment analysis task, we use the word list provided by Liu et al. [15]. We had 1,856 words in each class. For hate speech detection, we used the list of hate speech words from here[2]. For non-hate words, we used POS-tagging to get positive adjectives and adverbs. We also collect neutral words from Wikipedia. There were 426 words in each class. For formality detection, we used the list provided by Julian Brooke along with words collected from the internet to get a final list with 620 words in each class. For humor detection, we used the word list provided by Engelthaler and Hills [9] to get 400 words in each class. For NER, we used the same list as Formality detection.

4 Results and Discussion

Tables 2 show the results for different downstream tasks. We see that tailoring the BERT model using selective masking methodology helps and it outperforms BERT's random masking for 4 out of 5 tasks. As expected, there were no gains in results for Humor Detection because we did not use a suitable word list. Thus, this shows that if there is a mismatch in the words found in the word list and the downstream task, our strategy will provide no gains. Due to architectural limitations, we could not train BERT from scratch using the proposed methodology or try Gu et al. [10] like three-stage training. However, with just 20k steps, we are able to achieve improvements over random masking, showing the strength of the approach.

We notice that no single masking function performs the best for all the tasks. Masking Function 2 and 3 outperform Random Masking almost all the time. Masking Function 3 achieves the best results 70% of the time, suggesting the superiority of the exponential functions in capturing the relationship between masking probability and score. For no task, Masking Function 1 performed the best. This confirms the hypothesis of Liu [14] and Pavlick and Tetreault [21]. They argued that sentiments or formality does not always depend on extreme words, and we show it experimentally. To this end, our approach of unsupervised training using word lists and embeddings helps in capturing the domain-specific and task-specific patterns. Which masking function will perform the best depends upon the choice of the word list and downstream task.

Comparing our results with Gu et al. [10], they report a gain of 2.14% (compared to the BERT model) on the Movie Reviews dataset using selective masking and Amazon reviews dataset (in-domain data). We have achieved a gain of 1.94% (compared to the BERT model) in the accuracy using selective masking alone. When we use in-domain data too, we achieve a gain of 3.99% compared to the BERT model. Thus our strategy is effective. Gu et al. [10] cannot work for non-classification tasks. We showed that our approach works for NER as well and thus is more generalizable.

5 Conclusion

We presented a framework for tailoring the BERT model for downstream tasks using selective masking. Unlike the previous approach, we use easily accessible word lists instead of the specific in-domain data. Our approach is also more generalizable as it works for non-classification tasks as well. Experimental results on various downstream tasks show that our approach of training BERT using selective masking helps in capturing the domain-specific and task-specific patterns. But the

[2] https://www.cs.cmu.edu/~biglou/resources/bad-words.txt

method described in this paper cannot be generalized to all Natural Language Understanding (NLU) tasks. It would be an interesting problem to find a way to incorporate selective masking for other NLU tasks such as generic question answering or translation where there are no clear boundaries for categorizing/scoring the relevance of the words for the task at hand.

References

[1] Issa Annamoradnejad. 2021. Colbert: Using bert sentence embedding for humor detection.

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text.

[3] Souradip Chakraborty, Ekaba Bisong, Shweta Bhatt, Thomas Wagner, Riley Elliott, and Francesco Mosconi. 2020. BioMedBERT: A pre-trained biomedical language model for QA and IR. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 669–679, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[4] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

[5] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

[6] Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

[7] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

[8] 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.

[9] Tomas Engelthaler and Thomas T. Hills. 2018. Humor norms for 4,997 english words. *Behavior Research Methods*, 50(3):1116–1124.

[10] Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training.

[11] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

[12] Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *CoRR*, abs/1602.01585.

[13] Kexin Huang, Abhishek Singh, Sitong Chen, Edward Moseley, Chih-Ying Deng, Naomi George, and Charolotta Lindvall. 2020. Clinical XLNet: Modeling sequential clinical notes and predicting prolonged mechanical ventilation. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 94–100, Online. Association for Computational Linguistics.

[14] B. Liu. 2010. *Sentiment analysis and subjectivity*, pages 627–666.

[15] Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 342–351. ACM.

[16] 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.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

[18] Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter.

[19] Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819, Copenhagen, Denmark. Association for Computational Linguistics.

[20] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

[21] Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74.

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[23] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth M. Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. *CoRR*, abs/1909.04251.

[24] Sudha Rao and Joel R. Tetreault. 2018. Dear sir or madam, may I introduce the YAFC corpus: Corpus, benchmarks and metrics for formality style transfer. *CoRR*, abs/1803.06535.

[25] Wilson L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

[26] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

[27] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

[28] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.