Optimal masking strategies for masked language modelling and continuous pertaining for sentiment analysis

Matthew Koton (ID. 806614), Gavriel Cohen (ID. 339782344)

Submitted as final project report for the NLP course, Reichman, 2024

1 Introduction

Masked Language Modeling (MLM) has become a fundamental technique in Natural Language Processing (NLP), especially for pre-training large-scale language models (LLMs) like BERT and its variants [1]. MLM involves predicting masked words within a sentence, allowing models to learn rich contextual representations of language. Traditional methods use a fixed masking ratio and randomly select tokens for masking throughout the entire training process. In this paper, we explore alternative masking strategies to assess their impact on model performance.

Our research further extends to fine-tuning these various models, trained with different masking strategies, for sequence classification tasks - namely, sentiment analysis. By evaluating the performance across these tasks, we aim to identify if specific masking strategies during pre-training can lead to superior results in downstream applications.

This study is motivated by the rapid advancements in NLP and the growing demand for efficient training techniques. As LLMs increase in size, their training requires substantial computational resources. By exploring alternative masking strategies, we aim to optimize resource usage, potentially reducing training time and costs while maintaining or improving performance.

Problem Statements

1. Investigate novel masking techniques and evaluate their performance in comparison to traditional masking methods.
2. Discover which masking techniques deliver the best performance results.
3. Evaluate whether these potential performance increases are extendable to downstream tasks - for the purposes of this paper, sentiment analysis.

1.1 Related Works

Learning Better Masking for Better Language Model Pre-training [3] The authors argue that the traditional static approach to MLM may not yield optimal results, as the training needs of a model evolve over time. To address this, they propose two time-variant strategies: Masking Ratio Decay (MRD) and POS-Tagging Weighted (PTW) Masking. MRD adjusts the masking ratio dynamically during training, starting with a higher ratio and gradually decreasing it, which they found enhances the model's performance in downstream tasks. PTW Masking modifies the probability of masking different types of words based on their part-of-speech, allowing the model to focus more on 'difficult' words. These approaches lead to more efficient and effective pre-training, significantly improving the performance of models on various NLP tasks.

2 Solution

2.1 General approach

We decided on implementing 9 different masking strategies which we group into 2 groups, whole-word masking strategies and token-based masking strategies. Each masking strategy was implemented into its own data collator class for fine-tuning our models. Whole word masking ensured that if a word is made up of multiple tokens and a token within that word was masked, then the all tokens within that word would be masked as well. Token based masking does not have this requirement and allows for some tokens to be masked within a word while others are not.

Token-Based Strategies

1. Pseudo-max perplexity token based masking - we apply n random masks on each data chunk in the batch. We choose the mask that maximizes the perplexity of the chunk.
2. Pseudo-min perplexity token based masking - we apply n random masks on each data chunk in the batch. We choose the mask that minimizes the perplexity of the chunk.
3. Max perplexity token based masking - we calculate the per-token loss for every token in the chunk, and mask the topk highest perplexity tokens.
4. Min perplexity token based masking - we calculate the per-token loss for every token in the chunk, and mask the topk lowest perplexity tokens.
5. Random token based masking - the traditional random mask.

Whole-Word Strategies

1. Pseudo-max perplexity whole word masking - whole word variant of the pseudo-max strategy

2. Pseudo-min perplexity whole word masking - whole word variant of the pseudo-min strategy

3. Random whole word masking - whole word variant of the traditional random mask.

4. Part-of-speech (POS) token-based masking - Each sequence was passed through a pre-trained POS tagger model for tagging. We then mask the adjectives and verbs found in the data chunk.

Using these masking strategies, we trained various MLMs and recorded their results on the evaluation dataset. Additionally we took these models, froze the base model and fine-tuned their classification head for sequence analysis. We reported the fine tuned models results in this paper on classifying the sentiment of IMDB ratings.

2.2 Design

Base Model
We used the DistilBERT model, downloaded from the Hugging Face 'transformers' library as the base model for this study. DistilBERT is a suitable choice due to our computational power restraints as it is around 40% smaller than BERT, while retaining around 97% language proficiency [2].

Dataset
We used a downsampled IMDB dataset from the Hugging Face 'datasets' library. This is a suitable choice as it is a widely recognized benchmark for a wide range of NLP tasks. The dataset contains 50,000 highly polar movie reviews, evenly split between positive and negative sentiments. This provides a balanced and comprehensive foundation for training and evaluating our model. We used a downsampled version of this dataset, containing 10,000 samples each for the train and evaluation dataset.

Method
Our training data was fed through various data collators before being fed to the model for training. The masking took place within the data collator. Each masking strategy mentioned above was implemented through a separate data collator which was passed to our custom trainer class.

We kept the proportion of tokens or words to mask at $p = 0.15$. For the pseudo max and min strategies, we used $n = 10$. Each model was trained for 5 epochs due to resource and time constraints. Each model took around 30 minutes to train. The traditional random token-based or whole-word masking data collators were used over the entire evaluation dataset to allow for comparison between the two groups of models, respectively.

3 Experimental results

3.1 Fine-Tuning with Different Masking Strategies

Individual-Token Masking Schemes

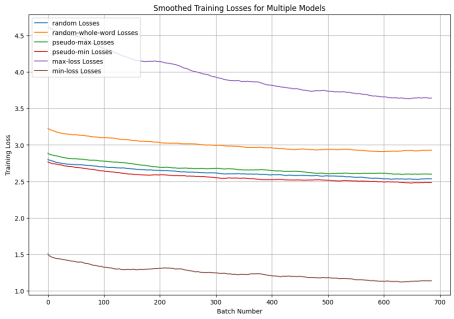

Figure 1: Smoothed Training Loss For the Token-Based Models

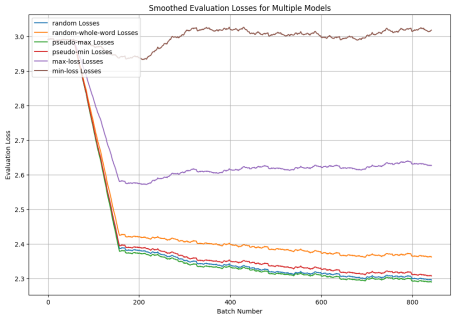

Figure 2: Smoothed Evaluation Loss For the Token-Based Models

In figure (1) and (2), we have plotted the losses for each token-based model on the masked training and evaluation set, respectively. The graphs are smoothed by taking the moving average of every 100 batches, so that the trends are more clearly visible. The traditional random whole-word masked model is plotted as well, for evaluating its performance against the token-based models.
The max-loss masking strategy exhibited the highest training loss and second-highest evaluation loss, which indicates it may be over-fitting to the rarer, more 'surprising' tokens, and therefore performs poorly across both dataset splits. Inversely, masking the lowest loss tokens during training resulted in the lowest training loss but high evaluation loss, indicating that the model is not learning effectively. This suggests that the model is under-fitting the training data by not being challenged by the more 'surprising' tokens in the fine-tuning dataset. The pseudo-max model had the lowest training loss, which indicates that it is a promising strategy for efficient fine-tuning.

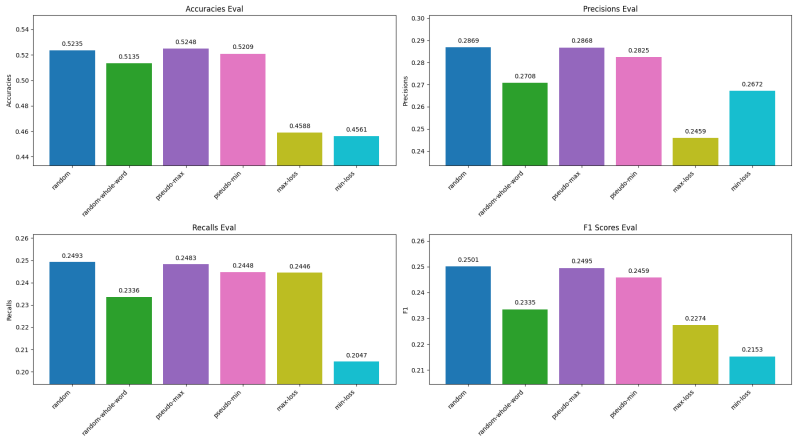

Figure 3: Accuracy, Precision, Recall and F1 Scores for the Token-Based Models

In figure (3) we see that the metrics for the pseudo and random strategies outperform the min and max loss strategies. The metrics for the random and pseudo-max models are on-par with each other. Both pseudo models outrank the traditional whole-word masking approach.

Whole-Word Masking Schemes

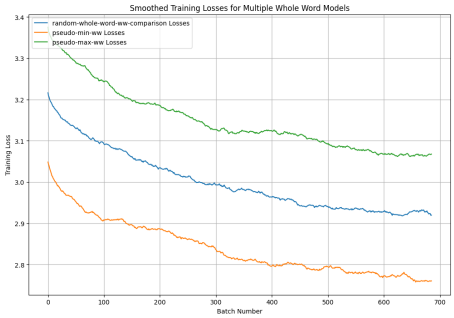

Figure 4: Smoothed Training Loss For the Whole-Word Models

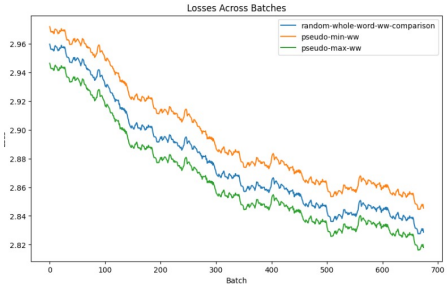

Figure 5: Smoothed Evaluation Loss For the Whole-Word Models

In figure (4) and (5), we have plotted the losses for each whole-word model on the masked training and evaluation set, respectively. The graphs are smoothed by taking the moving average of every 100 batches, so that the trends are more clearly visible. Due to computational constraints, we could not assess the POS tagger in this same round of tests.
The random whole-word masking performed second best in both training and evaluation loss. Note that whole-word pseudo max has the highest training loss, but the lowest evaluation loss. This indicates that this model effectively learns the underlying trends in the dataset better than traditional whole-word masking.

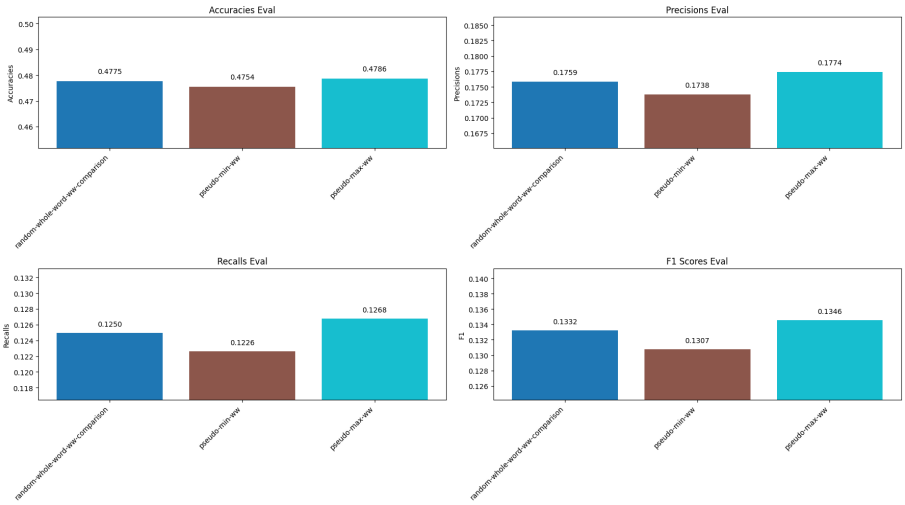

Figure 6: Accuracy, Precision, Recall and F1 Scores for the Whole-Word Models

Further, in figure (6) we can see that pseudo-max whole-word masking outperforms traditional random whole-word masking in accuracy, precision, recall and F1 score.

3.2 Sequence classification

Model Performance Comparison On Sentiment Analysis

	Base Model Masking Strategy	Loss (Epoch 1)	Loss (Epoch 2)	Accuracy (Epoch 1)	Accuracy (Epoch 2)
1	POS Masking	0.461	0.4182	0.8328	0.842
2	Pseudo Max	0.48	0.4365	0.831	0.839
3	Pseudo Min	0.4868	0.4433	0.829	0.835
4	Random Masking	0.481	0.438	0.8266	0.8326

Due to computational power and time restraints, sentiment analysis pre-training was performed for only the whole-word strategies. In sentiment analysis we achieved the best results when masking adjectives and verbs with the POS Masking strategy. Intuitively this makes sense as the model is learning to predict what an adjective will be based on the surrounding sentence. We can therefore infer that there is might be a relation between predicting what an adjective would be eg: the "good" dog or the "bad" dog and classifying sequences based on sentiment.

The next best performing strategy is pseudo-max whole word masking, which further strengthens the notion that this approach shows promise as being more efficient for whole-word masking compared to the traditional randomly applied whole-word masking.

4 Discussion

4.1 Limitations

The most challenging aspect of this study was the limitations on resources, as training large language models requires a lot of computational power. A byproduct of this limitation is the length of time it takes to train the models, which also proved challenging. We therefore needed to restrict certain parameters to ensure we could run the code in a reasonable amount of time. In the future, we would train more epochs so that we can more accurately compare these strategies.

4.2 Conclusion

In this study, we explored the impact of different masking strategies on the performance of masked language models. Our experiments revealed several key insights that contribute to our understanding of how these models learn and generalize from their training data.

Firstly, we found that pseudo-max whole-word masking outperformed the traditional random whole-word masking strategy. This result suggests that selectively masking words based on certain criteria—such as those that contribute most to the model's loss—can lead to more effective learning.

Furthermore, when evaluating the pre-trained models on downstream tasks such as sentiment analysis, we discovered that the model pre-trained with a focus on masking adjectives and verbs achieved the best performance. This outcome is particularly noteworthy as it highlights the importance of these parts of speech in understanding sentiment. Adjectives and verbs are often central to expressing opinions and emotions in text, thus pre-training a model with a focus on these elements may enhance its ability to discern the nuances of positive and negative sentiments in sentences. This insight aligns with the theory that targeted masking strategies can help models learn more specific linguistic properties, which in turn improve their performance on related tasks.

The results from our experiments underscore the importance of thoughtful masking strategies in the training phase of language models. While traditional random masking strategies are useful, they may not fully exploit the potential for more nuanced learning that targeted masking strategies can offer. The improved performance of models trained with pseudo max whole word masking and adjective-verb masking suggests that these approaches help the model learn more quickly and allow us to make better use of our training data.

5 Code

You can find the code used for our experiments and analysis in our GitHub repository, which is available at this link.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL `https://arxiv.org/abs/1810.04805`.

[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL `https://arxiv.org/abs/1910.01108`.

[3] Dongjie Yang, Zhuosheng Zhang, and Hai Zhao. Learning better masking for better language model pre-training, 2023. URL `https://arxiv.org/abs/2208.10806`.