NB-MLM: Efficient Domain Adaptation of Masked Language Models for Sentiment Analysis

Nikolay Arefyev[◇,▽,△] **Dmitrii Kharchev**[◇,▽] **Artem Shelmanov**[○,□,▽]
[◇]Samsung Research Center Russia / Moscow, Russia
[▽]Lomonosov Moscow State University / Moscow, Russia
[△]National Research University Higher School of Economics / Moscow, Russia
[○]Artificial Intelligence Research Institute / Moscow, Russia
[□]Sber AI Lab / Moscow, Russia
{nick.arefyev,dimitriy.kharchev}@gmail.com shelmanov@airi.net

Abstract

While Masked Language Models (MLM) are pre-trained on massive datasets, the additional training with the MLM objective on domain or task-specific data before fine-tuning for the final task is known to improve the final performance. This is usually referred to as the domain or task adaptation step. However, unlike the initial pre-training, this step is performed for each domain or task individually and is still rather slow, requiring several GPU days compared to several GPU hours required for the final task fine-tuning.

We argue that the standard MLM objective leads to inefficiency when it is used for the adaptation step because it mostly learns to predict the most frequent words, which are not necessarily related to a final task. We propose a technique for more efficient adaptation that focuses on predicting words with large weights of the Naive Bayes classifier trained for the task at hand, which are likely more relevant than the most frequent words. The proposed method provides faster adaptation and better final performance for sentiment analysis compared to the standard approach.

1 Introduction

Pre-training of neural networks with a language model (LM) or masked language model (MLM) objective on large amounts of non-domain-specific texts has given a significant boost of performance in almost all natural language processing tasks. While 16GB of texts were shown to BERT (Devlin et al., 2019) and ten times more to RoBERTa (Liu et al., 2019) during pre-training, the further training of these models with the MLM objective on domain-specific texts before fine-tuning to the target task was shown to further improve the final results (Sun et al., 2019; Gururangan et al., 2020). This technique is called the domain or task adaptation, depending on the degree of similarity of the data for adaptation to the target dataset. While initial pre-training is extremely expensive, it does not depend on the final task and can be performed only once. However, domain or task adaptation is done for each domain or task individually and is still quite resource-demanding, requiring hundreds of thousands of training steps or several GPU days, unlike final fine-tuning, which can often be done in a few GPU hours (Sun et al., 2019).

In this work, we propose a method for more efficient MLM adaptation. We have noticed that the standard MLM spends most of the training time on learning to restore the most frequent words like determiners or auxiliary verbs hidden (masked) from its input. While such training examples may be useful for learning English grammar, their domination during the adaptation phase seems to be wasteful for many final tasks. Since the final task and the dataset are already known in this phase, we propose to undersample such examples in favor of examples with targets related to the final task. This relatedness is estimated using a Naive Bayes classifier. Hence, we call our modified objective Naive Bayes Masked Language Model (NB-MLM). We hypothesize that hiding from the model and asking it to restore mostly features that are important for the final task will likely result in faster adaptation. Additionally, the absence of simple features and the requirement to restore them may teach the model to exploit more sophisticated and implicit features relevant to the final task.

We evaluate the proposed method on two datasets for sentiment analysis. It is one of the most popular tasks in natural language processing (Feldman, 2013) and an excellent playground for the comparison of adaptation methods due to the large amount of labeled and unlabeled user reviews of different products available. In particular, we consider the task of classifying the binary sentiment polarity of a given review. Our experiments

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9114–9124
November 7–11, 2021.

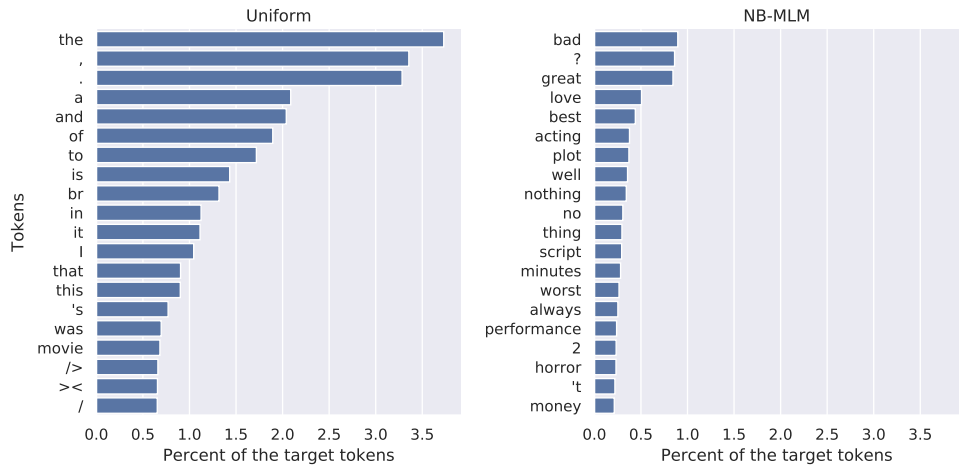

Figure 1: Target tokens that the model is asked to predict most often for Uniform MLM and NB-MLM.

show that the NB-MLM objective can significantly reduce adaptation time while achieving the same final performance or help to improve performance given the same amount of time for adaptation. [1]

2 Related Work

Pre-training Transformer networks with the MLM objective is proposed in (Devlin et al., 2019) for the BERT model and is shown to outperform the more traditional LM objective, though the similar task of predicting a word from its left and right context was used with different architectures earlier (Mikolov et al., 2013; Melamud et al., 2016). RoBERTa enhances BERT by pre-training longer on ten times larger corpora, getting rid of the next sentence prediction (NSP) task during pre-training, and selecting different target words to be masked and predicted in each epoch (dynamic masking).

Various approaches to further pre-training of BERT on domain or task-specific data are compared in (Sun et al., 2019), while Gururangan et al. (2020) carry out a similar investigation with RoBERTa. They try various options of data sources for adaptation: texts only from the target dataset (called task adaptation or within-task pre-training), larger datasets from the same domain (called domain adaptation or in-domain pre-training), and datasets from different domains (called cross-domain pre-training). They find the task adaptation, which is a computationally cheapest option, to be a surprisingly good solution. In their experiments, it often outperforms the domain adaptation and is only marginally worse than combining both methods. However, due to the large amount of data used in domain adaptation, Gururangan et al. (2020) train the MLM only for one or very few epochs. We find that our method leveraging large data more efficiently makes the domain adaptation comparable to the task adaptation, and their combination is significantly better than each of them.

Our idea of employing Naive Bayes weights is inspired by the NB-SVM model (Wang and Manning, 2012; Mesnil et al., 2014), which scales bag-of-ngrams vectors with Naive Bayes classifier weights and then trains linear SVM or logistic regression classifiers on them. It proved to be a very strong baseline, often outperforming both linear and more sophisticated models from that time.

3 MLM Objectives for Adaptation

Uniform MLM. For each input example, the standard MLM objective, as proposed by Devlin et al. (2019), samples 15% of the input positions (subwords) for calculating the loss. The positions are sampled from the uniform distribution without replacement: $P(pos) \propto 1$. Then 80% of the tokens on sampled positions are masked (replaced with a [MASK] token), 10% are replaced with some random tokens from the uniform distribution over the vocabulary, and 10% are left intact.

NB-MLM. As an alternative, we propose sampling 15% of positions from a non-uniform distribution that gives higher probabilities to positions that contain subwords with high feature importance $fi(w)$: $P(pos) \propto exp(fi(w_{pos})/T)$, where the temperature T is the hyperparameter allowing to balance between uniform sampling and determin-

[1] The repository for this paper: https://github.com/nvanva/nb-mlm

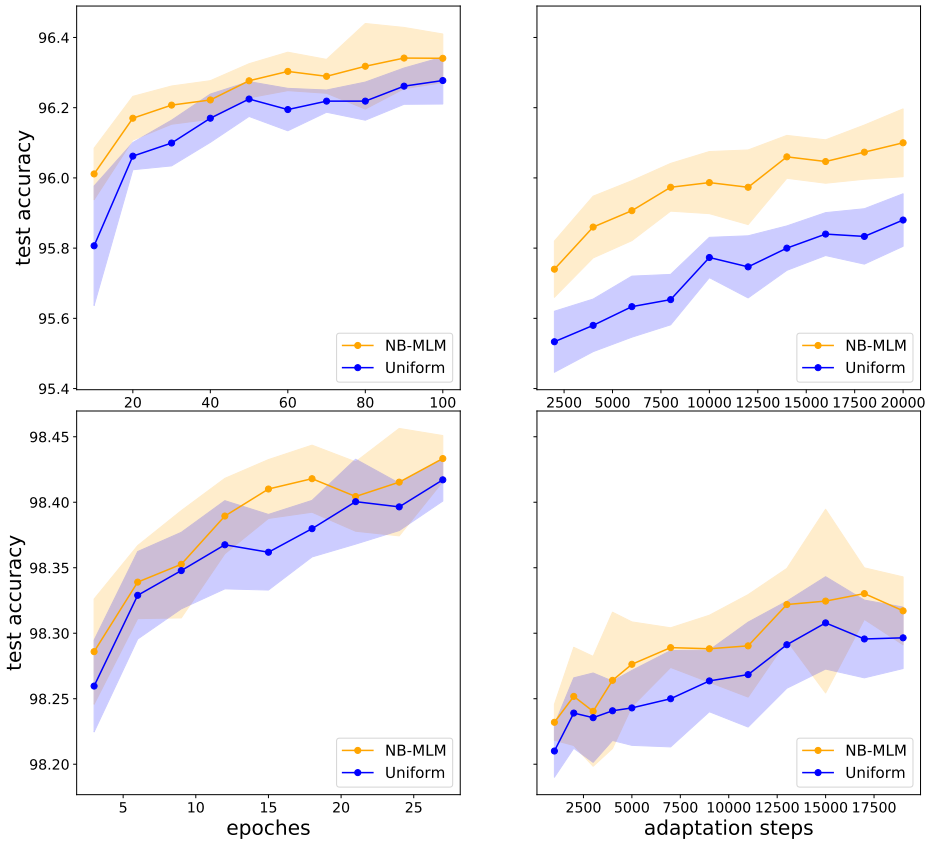

Figure 2: Task (left) and domain (right) adaptation with the standard Uniform MLM and the proposed NB-MLM objectives. Test accuracy on IMDB (top) and Yelp (bottom) for the classifiers fine-tuned from different MLM checkpoints saved during adaptation. Means and standard deviations over 6 runs are plotted for each model, except for DAPT on IMDB, where there were 15 runs. Corresponding dev accuracies are in Appendix B.

istic selection of positions that contain the most important features. For binary classification, the feature importance is estimated using the Naive Bayes classifier weights as follows:

$$fi(w) = |logP(w|1) - logP(w|0)|.$$

Thus, those features that are much more probable in one class than in another receive the highest scores. Similar to the method proposed by Wang and Manning (2012), the probabilities are estimated by the multinomial Naive Bayes model with additive smoothing ($alpha = 0.1$). Additionally, the scores are set to zero for those features that occurred in less than m examples to avoid the over-representation of unreliable features. As an example, Figure 1 shows the words that the model is most frequently asked to predict during the task adaptation on the IMDB movie reviews dataset ($T = 0.1$, $m = 5$ for NB-MLM). Evidently, NB-MLM learns to predict words relevant to sentiment analysis more often than the standard MLM.

Along with the uniform and NB-based distributions, during the preliminary experiments, we tried other options, which are described and compared in Appendix D. However, only NB-MLM outperformed the uniform baseline.

4 Experiments and Results

During the preliminary experiments described in Appendix A, we found that our method helps for both BERT and RoBERTa models. However, the latter model achieved significantly better performance. Therefore, we describe the results for RoBERTa in the rest of the paper.

For domain adaptation (denoted as DAPT), we employed the Amazon Reviews dataset (McAuley et al., 2015) with duplicates removed. We removed reviews shorter than 500 characters and split the rest into the training and validation sets of 21M and 10K reviews correspondingly. The valida-

Model	IMDB		Yelp P.	
	ERR	macro-F1	ERR	macro-F1
Our experiments with task and domain adaptation for RoBERTa-base				
Uniform short DAPT	4.42**	95.58**	1.78	98.22
NB-MLM short DAPT	4.14**	95.86**	1.77	98.23
Uniform DAPT	4.19**	95.81**	1.73	98.27
NB-MLM DAPT	3.85**	96.15**	1.71	98.29
Uniform short all-TAPT	3.92	96.08	1.70	98.30
NB-MLM short all-TAPT	3.82	96.18	1.69	98.31
Uniform all-TAPT	3.74	96.26	1.59	98.41
NB-MLM all-TAPT	3.66	96.34	1.58	98.42
Uniform short DAPT+all-TAPT	3.96*	96.04*	1.69	98.31
NB-MLM short DAPT+all-TAPT	3.73*	96.27*	1.66	98.34
Uniform DAPT+all-TAPT	3.62	96.38	1.55	98.45
NB-MLM DAPT+all-TAPT	**3.54**	**96.46**	**1.51**	**98.49**
Previously published results of the task and domain adaptation for BERT-base and RoBERTa-base				
BERT-base+ITPT (Sun et al., 2019)	4.37	-	1.92	-
BERT-base+IDPT (Sun et al., 2019)	4.88	-	1.87	-
RoBERTa-base+DAPT (Gururangan et al., 2020)	-	95.4	-	-
RoBERTa-base+TAPT (Gururangan et al., 2020)	-	95.5	-	-
RoBERTa-base+DAPT+TAPT (Gururangan et al., 2020)	-	95.6	-	-
RoBERTa-base+Curated-TAPT (Gururangan et al., 2020)	-	95.7	-	-
RoBERTa-base+DAPT+Curated-TAPT (Gururangan et al., 2020)	-	95.8	-	-
Large SOTA models (not directly comparable to our models)				
BERT-large+ITPT (Sun et al., 2019)	4.21	-	1.81	-
XLNET-large (Yang et al., 2019)	**3.20**	-	**1.37**	-

Table 1: Comparison of NB-MLM to the standard Uniform MLM and to the previously published results. From runs of each model with different random seeds, medians are taken. Adaptation scenarios where the difference between NB-MLM and Uniform MLM is statistically significant according to the McNemar's test are marked with * (p-value < 0.05) or ** (p-value < 0.01). ITPT (within-task pre-training), TAPT (task-adaptive pre-training), and Curated-TAPT (TAPT with extra unlabeled data from IMDB) denote further MLM pre-training on the target dataset only, which is similar to our all-TAPT. IDPT (in-domain pre-training) and DAPT (domain-adaptive pre-training) correspond to our DAPT. The best results for base and large models separately are in bold.

tion set was used to calculate perplexity during MLM training. For task adaptation (denoted as all-TAPT), we used all texts (without labels) from the target dataset, i.e. IMDB (Maas et al., 2011) or Yelp (Zhang et al., 2015) [2]. For IMDB, we employed the split of Gururangan et al. (2020) to make the results of our experiments directly comparable with their results. We used the binary classification version of Yelp (Zhang et al., 2015). For validation, we randomly selected 5K positive and 5K negative examples.

For domain and task adaptation, we used the batch size of 1024, while classifiers were fine-tuned with the batch size of 32. Based on our preliminary experiments, we set the learning rate of 2e-4 for the domain adaptation, 1e-4 for the task adaptation, and 1e-5 for final fine-tuning. Following Gururangan et al. (2020), we performed domain adaptation for one epoch on the Amazon dataset (20K steps, 38h on one V100 GPU) and task adaptation for 100 epochs on IMDB (18h) and 24 epochs on Yelp (14h). To show that NB-MLM can obtain results similar to Uniform MLM in a much shorter time, we also report the results of short adaptation with the duration reduced to 4K steps on Amazon, 20 epochs on IMDB, and 6 epochs on Yelp. To estimate the variance of the results due to the randomness in the order of training examples and positions selected for masking and prediction, we have trained each model with different random seeds. For both Uniform MLM and NB-MLM, we aggregated metrics from 15 runs for DAPT on IMDB, 3 runs for DAPT+all-TAPT on both IMDB and Yelp, and 6 runs for all other scenarios. The classifiers were fine-tuned for 4 epochs on IMDB and 2 epochs on Yelp [3]. For task adaptation with NB-MLM, we set $T = 0.4$, $m = 50$ based on preliminary experiments (see Appendix A). For domain adaptation with NB-MLM, we set $T = 0.1$, $m = 10$ on IMDB and $T = 0.1$, $m = 50$ after grid search from $T = [0.05, 0.1, 0.2, 0.4, 0.8]$, $m = [10, 50]$. Generally, for task adaptation with

[2] Using the whole target dataset for task adaptation has shown the best results for both Uniform MLM and NB-MLM, see Appendix C. This setup, when test examples (without labels) are exploited during training, is known as transductive learning.

[3] Longer fine-tuning resulted in a higher variance of metrics and worse final performance due to strong over-fitting.

many epochs of training on smaller datasets, larger temperatures are required to avoid over-fitting due to the same words masked in each example at each epoch. For domain adaptation, only one epoch of training is done on a large dataset, hence, smaller temperatures perform better.

Figure 2 shows how the final classification accuracy improves during the task and domain adaptation. Our NB-MLM model significantly helps for domain adaptation on IMDB. For task adaptation, the difference is much smaller and fits into two standard deviations. Still, on average, the NB-MLM seems to provide a consistent improvement throughout the adaptation. For Yelp, the improvements from NB-MLM are also small but consistent.

Table 1 compares our models and the previously published results on the test sets. In order to apply McNemar's test for statistical significance, instead of averaging across all runs of each model with different random seeds, we have to take predictions of a particular run. Thus, for each of our models, we selected the run with the median performance (for the even number of runs, the one just above the median) and reported its performance in the table.

For IMDB, the domain adaptation with NB-MLM obtains results similar to the Uniform MLM in 5x fewer training steps and data (only 20% of the data is seen during the first 4K steps). When trained for one epoch, it improves the results by more than 0.3%, which is also statistically significant. For task adaptation, the NB-MLM gives a much smaller improvement. Similarly to the results of Gururangan et al. (2020), in our experiments, the task adaptation with the Uniform MLM outperforms the domain adaptation that employs much more data by almost 0.5%. We suppose that this is due to the small proportion of relevant examples sampled by the Uniform MLM, which require many repetitions to learn from. Probably, training domain adaptation for hundreds of epochs, similarly to task adaptation, can fix this problem, but this is not feasible for large datasets and moderate computation resources. More efficient domain adaptation with NB-MLM, which focuses on targets that are likely relevant for the final task, reduces this difference to 0.2%. Finally, using the domain adaptation followed by the task adaptation results in the best final performance. In this scenario, NB-MLM gives 0.2% improvement for short adaptation and 0.1% for long adaptation. For Yelp, the metrics are higher, and the differences are smaller but still consistent.

5 Conclusion

We proposed a technique for the more efficient domain and task adaptation of MLMs. It is especially helpful for leveraging large data efficiently during the domain adaptation, resulting in significantly shorter adaptation time or better performance.

Acknowledgements

We are very grateful to our anonymous reviewers for insightful comments. The contribution of Nikolay Arefyev to the paper was partially made within the framework of the HSE University Basic Research Program. This research was supported in part through computational resources of HPC facilities at HSE University (Kostenetskiy et al., 2021).

References

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.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Commun. ACM*, 56:82–89.

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.

P. S. Kostenetskiy, R. A. Chulkevich, and V. I. Kozyrev. 2021. HPC resources of the higher school of economics. *Journal of Physics: Conference Series*, 1740:012050.

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*, abs/1907.11692.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of*

the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA. Association for Computing Machinery.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. Association for Computational Linguistics.

Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato, and Yoshua Bengio. 2014. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *ArXiv abs/1412.5335*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *Chinese Computational Linguistics - 18th China National Conference, CCL 2019, Kunming, China, October 18-20, 2019, Proceedings*, volume 11856 of *Lecture Notes in Computer Science*, pages 194–206. Springer.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. 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. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

A Preliminary Experiments with BERT and RoBERTa

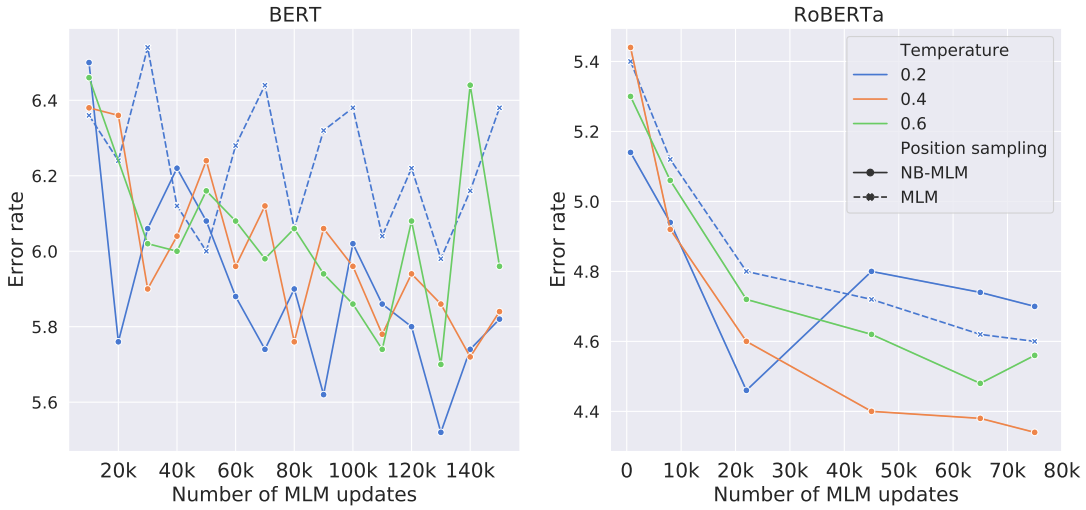

Figure 3: BERT ($m = 100$) and RoBERTa ($m = 50$) best error rates on the IMDB dev set from our split.

To verify our hypothesis, in the preliminary experiments, we tried improving the results of the ITPT (withIn-Task Pre-Training) method (Sun et al., 2019). Since no code for this paper was available at that time, we implemented this method using the Transformers library (Wolf et al., 2020)[4], which closely followed the details and hyperparameters specified in the paper but adopted recommendations from more recent models by not using NSP prediction and exploiting dynamic masking. Since no official development set is available for the IMDB dataset (Maas et al., 2011) and the split is not specified in the paper, for early stopping during classifier fine-tuning and NB-MLM hyperparameters selection, we employed our own split [5]. Note that this split was used only for preliminary experiments; later, we switched to the split of Gururangan et al. (2020). For adaptation, we used the whole dataset, excluding half of the development set to measure the validation perplexity.

Figure 3 (left) shows the final classification error rate depending on the number of adaptation steps. The best error rate on the development set across 10 epochs of the classifier fine-tuning is shown. Evidently, NB-MLM outperforms MLM on average. Despite the variance of their difference being rather high, we can see that after 60K adaptation steps, NB-MLM with the best temperature robustly shows equal or better results than the best result of MLM across 150K adaptation steps, which is almost 2.5x speedup. For comparison, Figure 3 (right) shows the results for RoBERTa using the same split. Evidently, RoBERTa with NB-MLM adaptation robustly outperforms MLM. For small temperature $T = 0.2$ after 20K steps of adaptation, we receive better results than MLM trained more than 3 times longer. However, later the performance drops significantly for the smallest temperature. Inspecting perplexity during adaptation, we found that the model begins to strongly overfit after 20K steps, which is likely related to the same positions for masking and prediction sampled at each epoch. Larger temperature $T = 0.4$ provides smaller benefits in the short run but gives more robust improvements and better final results. Overall, after 20K steps, it gives the same performance as the MLM trained for 75K steps, which is almost 4x speedup.

[4] https://github.com/huggingface/transformers
[5] https://github.com/nvanva/filimdb_evaluation/blob/master/FILIMDB.tar.gz

B Results on the Development Sets

Model	IMDB		Yelp P.	
	ERR	macro-F1	ERR	macro-F1
Our experiments with task and domain adaptation for RoBERTa-base				
Uniform short DAPT	4.3	95.7	1.6	98.4
NB-MLM short DAPT	3.7	96.3	1.5	98.5
Uniform DAPT	4.0	96.0	1.6	98.4
NB-MLM DAPT	3.5	96.5	1.6	98.4
Uniform short all-TAPT	3.6	96.4	1.5	98.5
NB-MLM short all-TAPT	3.7	96.3	1.5	98.5
Uniform all-TAPT	3.8	96.2	1.5	98.5
NB-MLM all-TAPT	3.5	96.5	1.4	98.6
Uniform short DAPT+all-TAPT	3.8	96.2	1.6	98.4
NB-MLM short DAPT+all-TAPT	3.5	96.5	1.6	98.4
Uniform DAPT+all-TAPT	3.4	96.6	1.5	98.5
NB-MLM DAPT+all-TAPT	3.4	96.6	1.5	98.5

Table 2: Validation metrics corresponding to the test metrics from Table 1 and used for early stopping. The results are rounded to one decimal place due to logging issues.

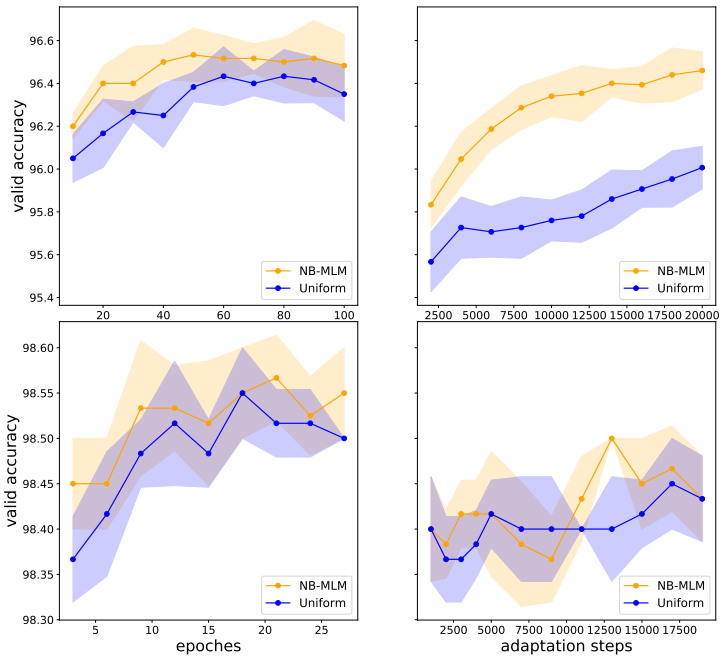

Figure 4: Task (left) and domain (right) adaptation with the standard Uniform MLM and the proposed NB-MLM objectives. Means and standard deviations over best dev accuracies on IMDB (top) and Yelp (bottom) corresponding to the test accuracies from Figure 2 and used for early stopping are shown. There were 15 runs for DAPT on IMDB and 6 runs for other models.

In this section, we show the results on the development sets corresponding to the results on the test sets provided in the main text. Since these results were used to select hyperparameters and also for early stopping during fine-tuning of the classifiers, they are less reliable to draw conclusions about final classification performance and shall be considered only together with the results on the test sets. While general trends are the same, we notice that for domain adaptation, the gap between NB-MLM and Uniform MLM on the IMDB dev set (figure 4, top right) is twice as large as on the test set. This may be due to the large variance of classification accuracy during fine-tuning and using early-stopping on the development set.

C Comparison of Various Subsets of IMDB for Task Adaptation

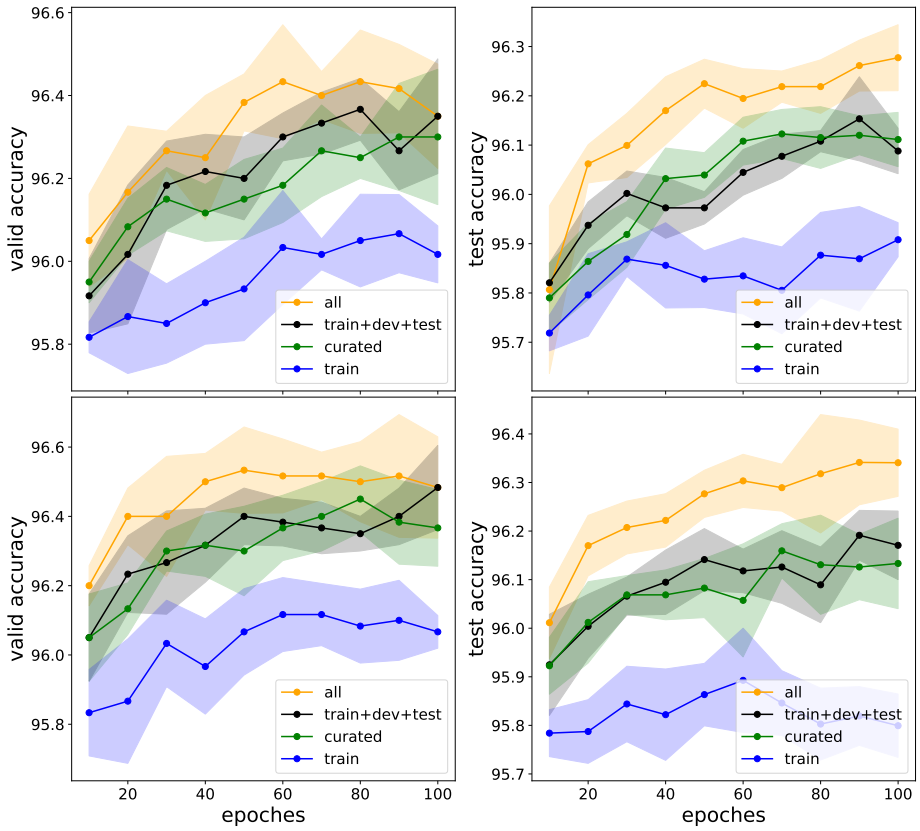

Figure 5: Task adaptation of Uniform MLM (top) and NB-MLM (bottom) on different subsets of IMDB. Accuracy on the IMDB dev (left) and test (right) sets. Means and standard deviations over 6 runs are shown for each model.

The all-TAPT method from our experiments employs examples (without labels) from all subsets of the target dataset for MLM training during task adaptation. For IMDB, this includes examples from the train (20K), unlabeled (50K), dev (5K), and test (25K) sets. We excluded only 1K examples from the unlabeled set in order to calculate the validation MLM loss on them. The scenario when test examples (without labels) are shown along with other examples during training is known as transductive learning. Alternatively, Gururangan et al. (2020) performs task adaptation on the train set alone or the concatenation of the train and the unlabeled sets. They denote the latter as Curated-TAPT.

Figure 5 compares these alternatives for Uniform MLM and NB-MLM. For NB-MLM, we selected optimal hyperparameters on the development set individually for each alternative, resulting in $T = 0.4, m = 50$ for all-TAPT and train+dev+test, and $T = 0.8, m = 50$ for other alternatives. In line with Gururangan et al. (2020), we see that additional examples from the unlabeled set significantly help for both models compared to adaptation on the train set only. Adding dev and test examples further improves their performance.

To understand if this improvement comes from simply adding more examples or adding exactly test examples, we additionally plot charts for task adaptation on train+dev+test subsets. Adaptation on train+dev+test (50K examples) is on par with Curated-TAPT (70k examples from the train and unlabeled subsets) while trained on approximately 1.5x fewer examples. However, adding the dev and test sets robustly improves Curated-TAPT while increasing the number of examples by the same factor. This probably indicates that using test examples for adaptation provides more benefits for performance than simply adding a comparable amount of other examples. Still, this question deserves more experiments and is out of the scope of this work.

D Alternatives for the Naive Bayes Weights

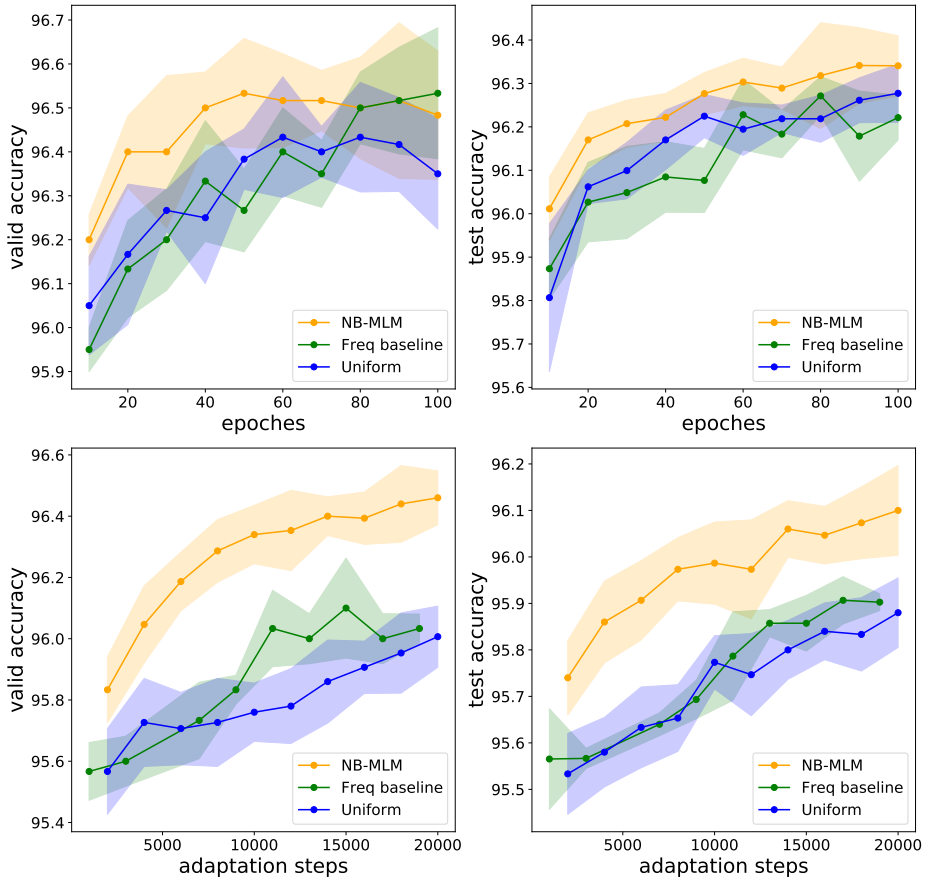

Figure 6: Comparison with the frequency-based baseline on the IMDB dev (left) and test (right) sets for all-TAPT (top) and DAPT (bottom).

The uniform distribution over positions is traditionally used to sample target subwords that are masked and predicted during MLM pre-training and adaptation. However, as Figure 1 shows, it makes the model learning to predict mostly frequent functional words such as articles, prepositions, pronouns, etc. While it may teach the model to extract some grammar-related features perfectly, it may also prevent the model from learning more specific features required for the final task due to rare necessity in such features during MLM training and limited model capacity. To solve this problem, we may simply lower the probability of sampling positions containing frequent words. Figure 6 compares the standard Uniform MLM and the proposed NB-MLM to a frequency-based baseline. In this baseline, we perform domain or task adaptation similarly to NB-MLM, but sample positions from $P(pos) \propto (\frac{1}{freq(w_{pos})})^{\frac{1}{n}}$, where n plays the same role as the temperature in NB-MLM, allowing to balance between sampling positions from the uniform distribution and selecting positions containing the most infrequent words. Word frequencies $freq(w)$ are estimated from the training subset of the IMDB dataset. We selected optimal values of n on the IMDB development set for all-TAPT and DAPT separately, resulting in $n = 3.5$ and $n = 2.5$ correspondingly. Evidently, the frequency-based baseline is on par with the uniform baseline. There are occasional improvements in the best validation accuracy, but they do not convert into improvements on the test set.

Next, we introduce another alternative, which is based on the conditional pointwise mutual informa-

tion between tokens and classes given context:

$$PMI(w, c|ctx) = \log\left(\frac{P(w|c, ctx)}{P(w|ctx)}\right).$$

Conceptually, it prefers to select tokens that are easier to predict based on the nearby context and class of the example than from the context alone. We supposed that learning to predict such tokens will make the model extract class-related features from the whole example rather than use only nearby context. We define the nearby context as one preceding token and one succeeding token and minimize PMI over them while maximizing it over classes. This means that we prefer selecting tokens, which are not easily predicted from either preceding or succeeding tokens but are much better predicted, at least for examples of one of the classes if that class is known.

$$fi(w_i) = \max_c \min_{ctx \in \{w_{i-1}, w_{i+1}\}} PMI(w_i, c|ctx)$$

Similarly to NB-MLM, we estimated these weights from the IMDB training set and set them to zero for those tokens that appear in less than m examples. Then we apply temperature softmax to convert these weights into a probability distribution over positions. We selected the hyperparameters on the development set, resulting in $T = 0.1, m = 10$.

Figure 7 shows that for all-TAPT on IMDB, the weights based on conditional MI do not help to improve the results of the Uniform MLM, unlike NB weights. Based on these results, we did not experiment with them for DAPT.

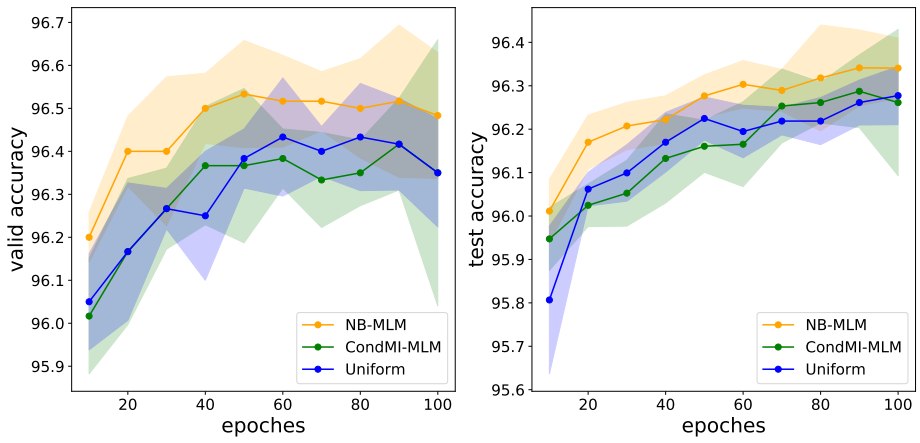

Figure 7: Comparison with the weights based on conditional mutual information for all-TAPT on the IMDB dev (left) and test (right) sets. Means and standard deviations over 6 runs are shown for each model.