SuryaKiran at PragTag 2023 - Benchmarking Domain Adaptation using Masked Language Modeling in Natural Language Processing For Specialized Data

Kunal Suri
Optum
kunal_suri@optum.com

Prakhar Mishra
Optum
prakhar_mishra29@optum.com

Albert Nanda
Optum
albert_nanda@optum.com

Abstract

Most transformer models are trained on English language corpus that contain text from forums like Wikipedia and Reddit. While these models are being used in many specialized domains such as scientific peer review, legal, and healthcare, their performance is subpar because they do not contain the information present in data relevant to such specialized domains. To help these models perform as well as possible on specialized domains, one of the approaches is to collect labeled data of that particular domain and fine-tune the transformer model of choice on such data. While a good approach, it suffers from the challenge of collecting a lot of labeled data which requires significant manual effort. Another way is to use unlabeled domain-specific data to pre-train these transformer model and then fine-tune this model on labeled data. We evaluate how transformer models perform when fine-tuned on labeled data after initial pre-training with unlabeled data. We compare their performance with a transformer model fine-tuned on labeled data without initial pre-training with unlabeled data. We perform this comparison on a dataset of Scientific Peer Reviews provided by organizers of PragTag-2023 Shared Task[1] and observe that a transformer model fine-tuned on labeled data after initial pre-training on unlabeled data using Masked Language Modelling outperforms a transformer model fine-tuned only on labeled data without initial pre-training with unlabeled data using Masked Language Modelling.

1 Introduction

Transformer based models like BERT Devlin et al. (2019), RoBERTa Liu et al. (2019), and DeBERTa He et al. (2020) have become de-facto models for Natural Language Processing (NLP) tasks outperforming all past techniques by significant margins. However, most of these models are originally trained on English corpus such as BookCorpus Yao and Huang (2018), English Wikipedia, and OpenWebText Liu et al. (2019). This becomes an issue when dealing with data from specialized domains such as medicine, healthcare, law, scientific peer reviews, etc. because these models are not aware of the specialized vocabulary in the domains due to which their performance is is generally subpar. This can be seen in Lee et al. (2019) where BERT performs poorly as compare to a model initialized with BERT weights and pre-trained on medical data. Training Transformer based models on data of specialized domain from the ground up poses significant challenges due to the scarcity of extensive datasets within these domains. So we resort to the practice of refining models originally trained on the English corpus by incorporating data sourced from such domains. Traditionally, this refinement process entails acquiring labeled data, structured according to well-defined formats pertinent to a task within the domain of interest. Subsequently, the model undergoes fine-tuning using this collected data. This approach is not efficient due to the labor-intensive and expensive nature of gathering substantial volume of labeled data. An alternative strategy – when we have a lot of unlabeled data and only a handful of labeled data - is domain adaptation (DA). In this paper we benchmark Masked Language Modelling (MLM) Devlin et al. (2019) as a DA strategy and see how it performs on PragTag-2023 Shared Task Dycke et al. (2023a). Although it is one of the strategies used to pre-train BERT, it has shown promise as a DA technique as can be seen in Ladkat et al. (2022), Karouzos et al. (2021).

2 Related Work

According to V7 Labs [2], Domain Adaptation (DA) is a technique to improve the performance of a model on a target domain containing insufficient

[1] https://www.aclweb.org/portal/content/pragtag-shared-task-argmining-workshop-2023

[2] https://www.v7labs.com/blog/domain-adaptation-guide

Proceedings of the 10th Workshop on Argument Mining, pages 218–222

annotated data by using the knowledge learned by the model from another related domain with adequate labeled data. Source Domain is the data distribution on which the model is trained using labeled examples. Target domain is the data distribution on which a model pre-trained on a different domain is used to perform a similar task. In this paper, Source Domain is the data distribution present in english corpus such as BookCorpus, English Wikipedia, and OpenWebText and Target Domain is the data distribution present in the data of this shared task.

There are primarily four types of DA techniques - Supervised DA, Semi-Supervised DA, Weakly Supervised DA, and Unsupervised DA. For this paper, we will primarily focus on Supervised and Unsupervised DA. In Supervised Domain Adaptation (SDA), target domain data is completely labeled. In Unsupervised Domain Adaptation (UDA), any kind of labels for the target domain data are entirely missing.

Lee et al. (2019) initialize BioBERT with weights from BERT, which was pre-trained on general domain corpora. Then, BioBERT is pre-trained on biomedical domain corpora. To show the effectiveness of our approach in biomedical text mining, BioBERT is fine-tuned and evaluated on three popular biomedical text mining tasks - NER, RE, and QA. The authors show that pre-training BERT on biomedical corpora largely improves its performance on these three tasks.

Karouzos et al. (2021) start from a model that is pretrained on general corpora, keep pretraining it on target domain data using the MLM task. On the final fine-tuning step, the update the model weights using both a classification loss on the labeled source data and Masked Language Modeling loss on the unlabeled target data.

Ladkat et al. (2022) use BERT-base model for MLM and finetune it for text classification on the target dataset. They freeze the encoder layer while training only the embedding and final task-specific dense layers. By doing so, they specialise the general domain word representations according to the target tasks and show that the performance of the resultant model is better than only BERT-base model.

In this paper, we will focus on Masked Language Modelling (MLM) which is a type of pre-training method that was introduced in BERT.

3 Task Description

In this task, we are given two datasets extracted from Kuznetsov et al. (2022). Both of these datasets contain a multi-domain collection of free-text peer reviews labeled with pragmatic labels on the sentence level. In the first dataset, each peer review can belong to medical articles, computer science, and scientific policy research. It has two parts parts - training dataset and test dataset. Training dataset is used to train the model and test dataset is used to evaluate the performance of the model trained on the training dataset. Going forward, we refer to these two datasets as *Train Dataset* and *Full Dataset* respectively. The second dataset is a secret test set *Secret Dataset*. Train, Full, and Secret Dataset contain same domains with Secret Dataset containing one additional domain not present in Train or Full Datasets. Every sentence in these datasets has one of the following pragmatic labels: Recap, Strength, Weakness, Todo, Other, and Structure. Our goal in this task is to correctly classify each peer review sentence into one of these categories.

In addition to these datasets, we use an auxiliary dataset, F1000raw, extracted from Dycke et al. (2023b) which is used for pre-training. This is a large unlabeled collection of peer reviews.

4 Methodology

For our experiments, we use DeBERTa-Base since it has been shown to perform better than models like BERT and RoBERTa. We first pre-train DeBERTa-Base on F1000raw using Masked Language Model (MLM) as shown in Fig.1. We then fine-tune this model using Train Dataset. We also fine-tune a DeBERTa-Base model using only Train Dataset without the pre-train step. We can see this workflow in Fig.2. We then pass each review from Full and Secret Datasets, take an average of the logits for all the classes and output the class with the highest logit score as shown in Fig.3. We then compare the performance of these two models and show that MLM helps improve the performance of the model on this classification task.

5 Implementation Details

Our solution comprises of two steps -

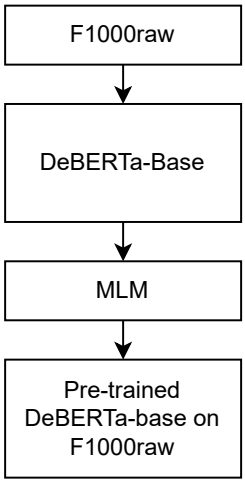

Figure 1: Pre-training on DeBERTa-base by using MLM Objective

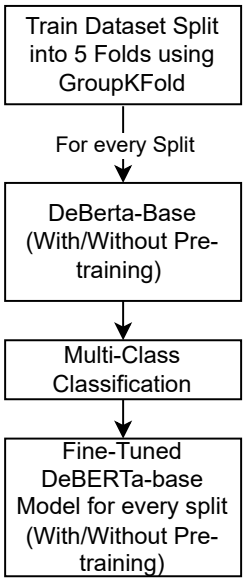

Figure 2: Finetuning DeBERTa-Base

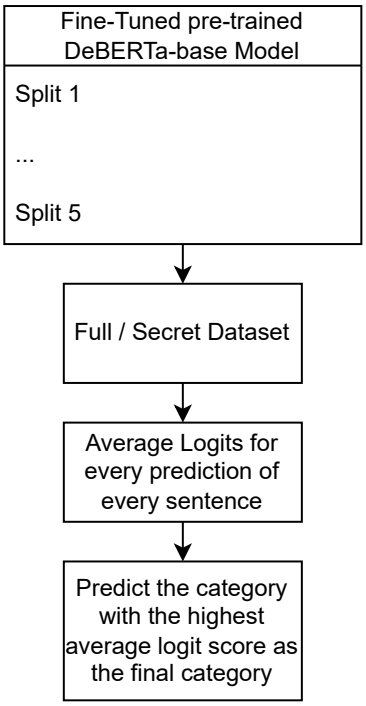

Figure 3: Inference on Fine-Tuned DeBERTA-base

5.1 Masked Language Modelling using F1000raw

As part of this step, we use all reviews in F1000raw. We combine these reviews and randomly split them into train, validation, and test datasets with 50%, 25%, and 25% share of data. We tokenize each of these datasets using a tokenizer created from the DeBERTa-base model. After tokenization, we concatenate all the sequences and split the concatenated sequences into shorter chunks of block_size of 512. We used this block_size because it covers all reviews and also it is short enough for T4 GPU.

5.2 Fine-tuning on Train Dataset

In this step, we use train a multi-class classification model on the Train Dataset. Since our objective is to compare performance of fine-tuning a multi-class classification model on a domain adapated model vs fine-tuning a multi-class classification model on a base model without domain adaptation, we perform the below steps twice - first for the domain adapted DeBERTa-Base obtained from above step and second for DeBERTa-Base without domain adaptation.

We use GroupKFold Cross Validation Strategy from scikit-learn Pedregosa et al. (2011) in order to ensure that each domain belongs to either train or validation or test split. We perform a 5 GroupKFold to create 5 Train-Validation splits of the

Domain	Split 1		Split 2		Split 3		Split 4		Split 5	
	W	WO	W	WO	W	WO	W	WO	W	WO
RPKG	-	-	**84.75**	80.79	**73.89**	66.76	**76.33**	74.13	**79.90**	70.06
CASE	69.17	**76.13**	**87.66**	82.57	**82.07**	81.27	**90.55**	88.27	**68.58**	63.96
SCIP	**84.97**	73.44	**75.73**	72.13	**62.05**	59.89	**91.01**	68.18	**75.**	61.32
ISCB	**93.27**	83.46	75.27	**77.58**	-	-	**84.04**	81.48	**79.84**	78.91
DISO	68.35	**80.98**	38.18	**50.**	86.28	**88.73**	**80.63**	72.04	**97.13**	91.11
Mean	**78.94**	78.50	72.32	**72.62**	**76.10**	74.16	**84.51**	76.82	**80.09**	73.07

Table 1: Comparison of F1 Scores for With (W) and Without (WO) MLM for all 5 Splits

Domain	With MLM	Without MLM
RPKG	82.75%	**84.06%**
CASE	81.97%	**82.94%**
SCIP	**86.45%**	85.04%
ISCB	**81.81%**	80.75%
DISO	**82.76%**	81.42%
Mean	**83.15%**	82.84%

Table 2: F1 Score for Full Dataset

Domain	With MLM	Without MLM
RPKG	82.75%	**84.06%**
CASE	81.97%	**82.94%**
SCIP	**86.45%**	85.04%
ISCB	**81.81%**	80.75%
DISO	**82.76%**	81.42%
SECRET	**77.93%**	73.21%
Mean	**82.28%**	81.24%

Table 3: F1 Score for Secret Dataset

training data. Within every split, we perform another GroupKFold split to divide the Validation into Validation and Test datasets. This ensures that we get Test score for every fold and use validation set exclusively for getting the best model.

6 Results and Discussion

We evaluate results on three datasets - 1) *Train Dataset*, 2) *Full Dataset*, and 3) *Secret Dataset*. For evaluating performance using Train Dataset, we use test dataset created in 5.2 of every split and pass it through the model trained using training data from that split. For evaluating performance on Full and Secret Datasets, we pass each review from these datasets through all five models, take an average of the logits for all the classes and output the class with the highest logit score.

Train Dataset gives us an idea about how both of the models compare across different splits and if one model is consistently better than the other model. Full Dataset contains similar domains as we have in Train Dataset but doesn't contain target variable. In Secret Dataset we have a new domain in addition to domains present in Full Dataset. The scores for every split of Train Dataset can be found Table 1, scores for Full Dataset can be found in Table 2, and the scores for Secret Dataset can be found in Table 3.

One interesting observation from Table 1, 2, and 3 is that the domain adaptation seems to be working on only for some domains and not others. This might be discouraging as it suggest that MLM only works sporadically but it is actually not the case. The reason why MLM works for some domains and not for others is due to difference in word distributions in different domains. Interested readers can refer to the analysis in the Supplementary Materials Section for detailed analysis of word frequencies of various domains in full and secret dataset and different splits of training data. The analysis shows us that domain adaptation is very effective in domains where the distribution has more words about Peer Reviews (which is the theme of this task) viz. SCIP, ISCB as compared to splits which have more health related terms viz. CASE and DISO.

7 Conclusion and future work

As we can see in the results, domain adapted DeBERTa-base beat DeBERTa-base without domain adaptation. While this is an encouraging result, how is this performance difference impacted by the scale of models and architecture of the model remains to be studied. We also need to study this problem on datasets from other niche domains as well. In addition to this, we can also study how domain adaptation impacts LLMs which are orders of magnitude larger than architectures such as BERT, RoBERTa, and DeBERTa.

Limitations

One of the biggest limitations of this analysis would be utilization of GPUs with more RAM as the size of the models scale. For example - We had to settle for DeBERTa-base because DeBERTa-large wouldn't fit in a GPU with 24 GB RAM. So,

as we analyse models with more parameters, we might have use GPUs with more RAM which might be a financial constraint to some teams.

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.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023a. Argmining 2023 shared task - pragtag: Low-resource multi-domain pragmatic tagging of peer reviews. In *Proceedings of the 10th Workshop on Argument Mining*, Singapore. Association for Computational Linguistics.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023b. NLPeer: A unified resource for the computational study of peer review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5049–5073, Toronto, Canada. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Constantinos Karouzos, Georgios Paraskevopoulos, and Alexandros Potamianos. 2021. UDALM: Unsupervised domain adaptation through language modeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2579–2590, Online. Association for Computational Linguistics.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and Resubmit: An Intertextual Model of Text-based Collaboration in Peer Review. *Computational Linguistics*, 48(4):949–986.

Arnav Ladkat, Aamir Miyajiwala, Samiksha Jagadale, Rekha A. Kulkarni, and Raviraj Joshi. 2022. Towards simple and efficient task-adaptive pre-training for text classification. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 320–325, Online only. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

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. *CoRR*, abs/1907.11692.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Wenlin Yao and Ruihong Huang. 2018. Temporal event knowledge acquisition via identifying narratives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 537–547, Melbourne, Australia. Association for Computational Linguistics.