Reducing Inference Time of Biomedical NER Tasks using Multi-Task Learning

Mukund Chaudhry[1] **Arman Kazmi**[1] **Akhilesh Verma**[1] **Vishal Samal** [1]
Shashank Jatav[1] **Kristopher Paul**[1] **Ashutosh Modi**[2]
Elucidata Inc., New Delhi[1] IIT Kanpur[2]
ashutoshm@cse.iitk.ac.in[2]
{mukund.chaudhry, arman.kazmi, akhilesh.verma, vishal.samal,
shashank.jatav, kristopher.paul } @elucidata.io[1]

Abstract

Recently, fine-tuned transformer-based models (e.g., PubMedBERT, BioBERT) have shown the state-of-the-art performance of several BioNLP tasks, such as Named Entity Recognition (NER). However, transformer-based models are complex, have millions of parameters, and are relatively slow during inference. In this paper, we address the time complexity limitations of the BioNLP transformer models. In particular, we propose a Multi-Task Learning based framework for jointly learning three different biomedical NER tasks. Our experiments show a reduction in inference time by a factor of three without any reduction in prediction accuracy.

1 Introduction

Transformer-based large language models (LLMs) have made it much easier to perform various NLP tasks with high accuracy. However, due to their large size, they take a lot of time and money to perform inference on large datasets. To give some perspective, one forward pass through PubMedBERT (Gu et al., 2020) takes 8-50ms on an AWS g4dn.xlarge instance [1] (which comes with an NVIDIA T4 GPU). Running one PubMedBERT model on 1 million biomedical paragraphs would take roughly 9 hours. Given the deluge of biological information daily, using fine-tuned PubMedBERT models for each biomedical NER task separately would be too time-consuming and expensive.

When it comes to deep learning models, there are generally two variables that are optimized before deployment. These are *size* (the space occupied by the model's weight on disk and RAM) and *inference time* (the time taken for one prediction).

Model size tends to matter more when deployed on edge devices and mobile phones since these devices have storage and RAM constraints. Several techniques, such as knowledge distillation, have been proposed to address this issue, and some of the prominent models which have achieved a significant decrease in model size without much decrease in accuracy are DistilBERT (Sanh et al., 2019), SqueezeBERT (Iandola et al., 2020), and MobileBERT (Sun et al., 2020). However, size is usually not an issue for models running on servers. For example, a PubMedBERT model has a size of only 400 MB. Instead, the main concern is inference time, which is what we focus on in this paper. Generally, a reduction in the model size naturally leads to a reduction in the inference time. However, in this work, we focus on reducing the inference time without reducing the model size.

Multi-Task Learning (MTL) primarily aims to improve the accuracy of multiple prediction tasks that are related to each other by leveraging commonly useful information. Many of the previous works have shown the effectiveness of multi-task learning-based models for BioNer tasks. The first work to apply MTL for biomedical named entities was attempted by Crichton et al. (2017). They used pre-trained word embeddings with CNN-based neural networks to extract named entities from biomedical texts. Wang et al. (2018) used a combination of BiLSTM and CRF-based model, adapted from Liu et al. (2018), to extract the entities and further used character and word-based embeddings that were shared by different datasets. A slightly different approach was proposed by Zuo and Zhang (2020), where they trained a dataset-aware MTL model and showed that their model was able to discriminatively exploit information from all of the related training datasets.

The recent developments of large language models, such as BERT (Devlin et al., 2018), have demonstrated the effectiveness of better contextualized representation of various NLP tasks. Lee et al. (2019) developed BioBERT using the BERT language model and pre-trained it on biomedical abstracts and papers. They achieved state-of-the-art

[1] https://aws.amazon.com/ec2/instance-types/g4/

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 116 - 122
December 15-18, 2022

results on several biomedical named entity recognition datasets. Khan et al. (2020) and Mehmood et al. (2019) incorporated MTL in BERT-based models and showed promising results to extract biomedical named entities.

Although the previous works have shown the importance of multi-task learning when incorporated with either neural network-based models or transformer-based models, none of them have targeted optimizing these large models. While deploying these models for prediction, inference time matters; hence, it is equally important to develop models that reduce the inference time without any significant drop in performance. To this end, we develop a multi-task learning model for three different entities (*cell-line*, *tissue*, and *strain*) and show that we can reduce the inference time by a factor of 3 without any drop in performance when compared with a single-task model for each entity.

Our main contributions are as follows:

- We fine-tune a multi-task PubMedBERT model, demonstrating a significant reduction in inference time.
- We compare the performance of our multi-task model with that of a single-task model and show that there is no significant drop in F1 scores. Further, we built a multi-class token classification model on our corpus and found that it performs the worst which shows the effectiveness of using a multi-task learning model.
- We release [2] a new gold-standard corpus manually tagged with *cell-line*, *tissue* and *strain* type entity, on which we report our results of the experiments performed. This dataset is the first of its kind that contains manual annotation of *tissue* and *strain* entities.

The rest of the paper is organized as follows. We provide the details of the dataset in section 2. The experiments, results and their analysis are shown in section 3 and 4 respectively. Finally, in section 5, we summarize all the results and provide pointers for future research.

2 Dataset

For the BioNER task, there are several publicly available annotated datasets but the most widely used datasets for benchmarking are JNLPBA (Collier and Kim, 2004; Huang et al., 2020), NCBI-Disease (Dogan et al., 2014), BC5CDR, (Li et al., 2016) BC2GM (Smith et al., 2008), and LINNEAUS (Gerner et al., 2010). These datasets cover mostly *cell line*, *cell type*, *chemical*, *disease*, *gene*, *protein*, and *species* type entities, and most of them rely on PubMed articles as a source. One of the significant concerns regarding most of the BioNER datasets is the data quality, which is not only limited to the biomedical domain. Li et al. (2022) mentioned annotation quality as one of the major challenges in the field of NER. An updated version of the 2004 JNLPBA challenge was released in 2019 to address the flaws in the original corpus (Collier and Kim, 2004; Huang et al., 2020). Another issue is the source and the entity type, which is generally targeted in these datasets. These benchmark datasets lack entities such as *tissue* and *strain* that can help create meaningful cohorts across experiments. This information can be used to control the genetic variability in datasets.

To address the issues mentioned above, we created a gold-standard corpus manually annotated with *cell-line*, *strain*, and *tissue* on abstracts extracted from the Gene Expression Omnibus (GEO) (Edgar et al., 2002) database. It is a public repository established by National Center for Biotech-

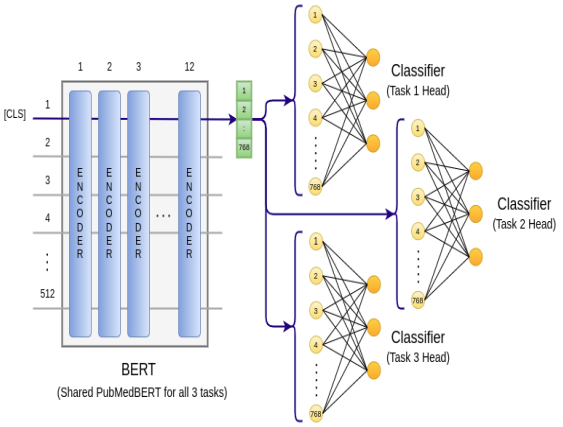

Figure 1: Architecture of the MTL model

Entity type	#Docs	#Words	#Mentions	#Unique Mentions	#Docs w/ at least one mention
strain	3560	234121	3476	574	2049
tissue	2607	430455	1804	338	961
cell-line	3059	532805	1541	483	677

Table 1: Summary statistics of the corpus (includes both the training and the test set.)

[2]The dataset and the source code of our experiments can be found here.

Entity	STL	MTL	Frozen1	Frozen2	Multi-class
Cell line	0.85	0.86	0.70	0.86	0.62
Tissue	0.71	0.71	0.52	0.01	0.46
Strain	0.88	0.87	0.63	0.61	-

Table 2: F1 scores of different models on each entity type.

nology Information (NCBI) for high-throughput gene expression data generated mainly through microarray technology. Several other data applications, such as those that look at genome methylation, chromatin structure, and genome-protein interactions, are now supported by GEO, which has developed along with the quickly changing technological landscape (Clough and Barrett, 2016).

The corpus was manually annotated by the domain experts, and the annotation guidelines followed can be found in Appendix A. The corpus consists of 9226 English paragraphs, and the number of mentions of *strain* (3476) is more than *cell line* (1541) and *tissue* (1804). Despite the less number of total mentions of *cell line*, the number of unique mentions of *cell line* (483) is far greater than the number of unique mentions of *tissue* (338). In the data, wherever the *strain* entity is tagged, the *cell line* and *tissue* are not found, and vice-versa. This is due to the nature of the abstracts (extracted from GEO) where we find either the texts contained mention of *cell line*, *tissue*, *strain* or both *cell line* and *tissue* in the same text. This makes the corpus unique and more reasonable to perform a multi-task learning model instead of building a multi-class token classification model. Table 1 provides more details of the corpus.

3 Experiments

In this section, we describe our experiments in detail about the model architecture, the training procedures, and the evaluation metrics followed.

3.1 MTL Model

Figure 1 shows the MTL architecture deployed in our work. The shared model follows the standard BERT architecture (Devlin et al., 2018) where the task heads consist of two linear layers. The first layer has a shape of 768 x 768, whose outputs are passed through the ReLU activation function and then fed into the second linear layer with a shape of 768 x 3. This layer acts as the token classifier, where each token is assigned one of three classes following the BIO tagging scheme.

3.1.1 Training and Evaluation Metrics

The training and testing split was 70:30. The shared model was initialized with PubMedBERT (Gu et al., 2020) weights, and the task heads were randomly initialized. We then fine-tuned the model for eight epochs at a learning rate of 2e-5 and a batch size of 20. Each batch consisted of examples from the three individual entities mentioned in different paragraphs. Each of the examples in the batch contributed to the loss of the task head for that particular example and to the shared BERT model.

To evaluate the model's performance, we consider each predicted entity as correct only if both the entity boundary and entity types are the same as the ground-truth annotation (i.e., exact match). We then calculate F1 scores for each entity type and report the results.

3.1.2 Controlling other factors

Different factors can affect BERT's inference time, such as batch size, sequence length, choice of deep-learning framework, and hardware. We used a batch size of 1 in all of our experiments, and to control the sequence length, we fixed the corpus that was used to test different model variants, ensuring it resembled production workloads. Regarding hardware, we used an AWS g4dn.xlarge [3] instance as our GPU machine and a laptop with Intel i5-7300U as our CPU machine. For all the experiments, we used Pytorch, [4].

3.2 Single Task Learning (STL) & Multi-class Token Classification Model

To compare the results of our multi-task learning model, we fine-tuned three different individual PubMedBERT models for *cell line*, *tissue*, and *strain* type entities. We refer to these models as single-task learning models as they are fine-tuned for each individual entity.

In general, for NER tasks, a multi-class token classification model is preferable. While in the case of biomedical text, all entities might not be mentioned in the same text; for example, in our case, the corpus did not have *strain* entity wherever there was mention of *tissue* and *cell line* entities. However, since *tissue* and *cell line* annotations were done together, it was possible to compare the results with that of the multi-task model. So, we

[3] https://aws.amazon.com/ec2/instance-types/g4/
[4] https://pytorch.org/

Model	Inference time (CPU)	Inference time (GPU)
Single-task (tissue + cell line + strain)	430 ± 16 ms	31 ± 1 ms
Multi-task	150 ± 6 ms	11 ± 1 ms

Table 3: The inference time per input (avg) of the MTL model compared to the single-task models run sequentially.

fine-tuned a multi-class token classification model combining the *tissue* and *cell line* paragraphs for eight epochs with a learning rate of 2e-5 and batch size of 16.

4 Results and Analysis

The results of our experiments are displayed in Table 2. The single-task learning model (STL) or the PubMedBERT model fine-tuned for three individual entities achieves an F1 score of 0.85, 0.71, and 0.88 for cell line, tissue, and strain, respectively. The third column shows the results of our MTL model fine-tuned jointly on the three tasks, and the F1 scores are 0.86, 0.71, and 0.87 for cell line, tissue, and strain, respectively which shows that there is no significant change in F1 score when compared to the single task model results. The MTL model for the cell line entity gives a better F1 score of 0.86 than the single-task learning model for the cell line. This shows that the MTL model is able to learn the mentions of cell line better than the other entities.

The results of the multi-class token classification model built over the paragraphs containing only cell line and tissue are 0.62 and 0.46, respectively. Since our data is unique in terms of the entities annotated and their mentions in the paragraphs, deploying a multi-class token classification model to learn the properties of the entities in the text is not a good choice in our case as it gives poor results.

It might be possible that the underlying PubMedBERT model learns the same features while fine-tuning for different NER tasks; hence, the MTL model is performing well. To rule out this possibility, we fine-tune the models after freezing the encoder layers of the PubMedBERT model. The fourth and fifth columns of Table 2 show the F1 scores when only the last layer with the task-specific head is trained during the fine-tuning process, and the underlying PubMedBERT layers are frozen. The Frozen1 model is initiated with the pre-trained PubMedBERT weights, and the Frozen2 model is initiated with the model's weights fine-tuned only on the cell line NER task. The F1 scores for the Frozen1 and Frozen2 models are quite poor, which clearly implies that jointly fine-tuning the MTL model on multiple NER tasks learns new features and performs better. The Frozen2 model achieves a good F1 score for the cell line because the underlying frozen model was fine-tuned for the same field.

4.1 3x reduction in inference time

Our primary finding is that an MTL model described above jointly trained on three different NER tasks gives the same model performance when compared to that of a PubMedBERT model fine-tuned separately for three tasks. Table 3 shows the average inference time taken by the MTL model and the single-task model when run sequentially. The MTL model takes around 11 ms on GPU and 150 ms on CPU, which is roughly three times less than the time taken by the single-task model. This shows the primary benefit of joint MTL training, which leads to a considerable reduction in inference time and cost and is crucial for practical applications. Instead of doing a forward pass through 3 separate BERT models to tag a paragraph of text, we only have to do it for one BERT model. The task heads themselves have a negligible contribution to inference time.

4.2 Low prediction accuracy for tissue

As seen in Table 2, the F1 scores for *tissue* field is much lower than that of *cell line* and *strain*. Even the single-task learning model fine-tuned for *tissue* entity gives an F1 score of 0.71 only. There might be two possible reasons for the poor performance. Firstly, *cell line* and *strain* names have a very different sub-word structure as compared to the *tissue* names and thus are significantly easier to detect. Secondly, detecting *tissue* names requires a deeper understanding of the surrounding context in which it occurs. For example, 'blood' can be a tissue, but it can also occur in a different context where it is not a tissue.

In order to see if we can improve the prediction accuracy for the tissue field, we fine-tuned an MTL model with two tasks. One was the actual NER task, and another was an auxiliary classification task that predicted whether an input paragraph had any tissue tag present or not. We tried several combinations of the learning rate, batch size, and weightage of the two tasks in the final loss func-

tion, but the best F1 score achieved was still 0.71, as reported in Table 2.

5 Conclusion and Future Work

In this study, we demonstrated how multi-task learning may be used to speed up model inference for complementary tasks that must be performed simultaneously on the same input. In particular, we compared our multi-task model to a single-task model and demonstrated that while the multi-task learning model's performance remained constant, the inference time was reduced by three. Moreover, for our experiments, we created a gold-standard corpus, manually tagged with *cell-line*, *tissue* and *strain*. This corpus is the first of its kind where three different entities are manually curated by domain experts.

When compared to the other entity types, the models' performance in identifying *tissue* names was incredibly poor, demonstrating how challenging it is to extract accurate *tissue* names from the text in the right context. For *tissue* NER, we must either discover a more suitable auxiliary task or develop some rule-based methods that will enhance the entity's overall performance. To increase the accuracy of *tissue*, we intend to carry out these actions in the future. Investigating the MTL model for inference time on benchmark datasets would be another interesting project.

Acknowledgements

We would like to thank the ICON-2022 anonymous reviewers and Shubhra Agrawal for their invaluable comments and feedback on this work. We also thank Amritanjali Kiran at Elucidata for helping in preparing the datasets.

References

Emily Clough and Tanya Barrett. 2016. The gene expression omnibus database. *Methods in molecular biology*, 1418:93–110.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

Gamal Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. 2017. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, 18.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Rezarta Dogan, Robert Leaman, and Zhiyong lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47.

Ron Edgar, Michael Domrachev, and Alex E. Lash. 2002. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30 1:207–10.

Martin Gerner, Goran Nenadic, and Casey Bergman. 2010. Linnaeus: A species name identification system for biomedical literature. *BMC bioinformatics*, 11:85.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Ming-Siang Huang, Po-Ting Lai, Pei-Yen Lin, Yu-Ting You, Richard Tzong-Han Tsai, and Wen-Lian Hsu. 2020. Biomedical named entity recognition and linking datasets: survey and our recent development. *Briefings in Bioinformatics*, 21(6):2219–2238.

Forrest N. Iandola, Albert E. Shaw, Ravi Krishna, and Kurt W. Keutzer. 2020. Squeezebert: What can computer vision teach nlp about efficient neural networks?

Muhammad Raza Khan, Morteza Ziyadi, and Mohamed A. Abdelhady. 2020. Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. *ArXiv*, abs/2001.08904.

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*.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016. Baw068.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Liyuan Liu, Jingbo Shang, Frank F. Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *AAAI Conference on Artificial Intelligence*.

Tahir Mehmood, Alfonso Gerevini, Alberto Lavelli, and Ivan Serina. 2019. *Leveraging Multi-task Learning for Biomedical Named Entity Recognition*, pages 431–444.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

L. Smith, L. Tanabe, R. Ando, C. Kuo, I-Fang Chung, C. Hsu, Y. Lin, Roman Klinger, Christoph Friedrich, K. Ganchev, M. Torii, Hongfang Liu, Barry Haddow, Craig Struble, Richard Povinelli, Andreas Vlachos, William Baumgartner Jr, Lawrence Hunter, B. Carpenter, and W. Wilbur. 2008. Overview of biocreative ii gene mention recognition. *Genome Biology*, 9.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2018. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752.

Mei Zuo and Yang Zhang. 2020. Dataset-aware multi-task learning approaches for biomedical named entity recognition. *Bioinformatics*, 36(15):4331–4338.

Appendix

A Manual Curation Guidelines

For annotating the corpus, two curators were recruited, and both had a biological background. The paragraphs were extracted based on the dataset ids from the GEO database and were exported to the Labelstudio [5] tool for annotations. Each dataset was assigned to the two curators for double-blinded curation where the curators curate the datasets assigned to them independently. The similarities were assessed for every dataset curated by two curators independently and in the case of dissimilarity, the dataset was passed to an expert curator for final annotations. Apart from this, about 10% of datasets were randomly picked for quality checks even if there was no dissimilarity.

The curation for *tissue* and *cell line* was done together and the ontology followed for *tissue* and *cell line* were the BRENDA Tissue Ontology (BTO) [6] and Cellosaurus (CVCL)[7] respectively. In the case of annotating *strain* entity, the strain of mouse and rats used during the experimental process was annotated. To find out the attribute of each mouse and rat provided in the experimental design of the dataset ids, the curators referred to Mouse Genome Informatics (MGI) [8] for the strain information.

B Dataset creation for Multi-class sequence model

Dataset for multi-class token classification model includes paragraphs with tag for *cell-line* (928), *tissue* (1347), *cell-line & tissue* (102), *none* (2557) which was split in 70:30 ratio for training and testing in stratified way.

[5] https://labelstud.io/
[6] https://www.ebi.ac.uk/ols/ontologies/bto
[7] https://www.cellosaurus.org/
[8] http://www.informatics.jax.org/home/strain