Literature Review: An Empirical Study of Multi-Task Learning in BERT for Biomedical Text Mining

Mateo Santiago Rueda Molano

Abstract

This paper corresponds to the literature review of the paper "An Empirical Study of Multi-Task Learning in BERT for Biomedical Text Mining" by Peng et al [6]. This paper proposes a multi-task model for different benchmarks in the tasks of text similarity, relation extraction, named entity recognition, and text inference. The proposed fine-tuned models outperform the state-of-the-art (BERT-based) models on four clinical and biomedical domains benchmarks. The importance of this work lies in proposing a single MTL model that researchers can use for different biomedical NLP tasks.

1. Introduction

Multi-task learning (MTL) is a subfield of Machine Learning in which multiple tasks are learned simultaneously by a shared model [1]. MTL is natural to use when we are interested in obtaining predictions for multiple tasks or when training data is scarce [8]. Models based on deep neural networks (like BERT), although robust, usually need millions of labeled samples to be trained [11]. In biomedical applications such as text mining, Multi-Task models become convenient compared to traditional DNN models since labeled samples are difficult to collect. Previously, MTL has been studied in biomedical and clinical NLP, but most studies focus on one task with multiple corpora [2] [9] or several tasks with a single corpus [10] [4].

To leverage this gap, the authors propose BERT-based Multi-Task (MT) models for the tasks proposed on the Biomedical Language Understanding Evaluation (BLUE) benchmark. This benchmark comprises eight tasks: text similarity, relation extraction, inference, and named entity recognition.

The paper's main contributions are: 1) An extensive experimentation on eight different benchmarks in biomedical and clinical texts. 2) The development of a fine-tuned MT model that outperforms BERT-based models on four benchmarks and has comparable performance on the others, considering the advantage of having a unique model.

2. Methods

The input *X* of the model can be one or many sentences separated by the token [*SEP*]. Similar to BERT, the input sequences X start with the token [*CLS*]. The MT-model comprises shared layers (based on the BERT model) and task-specific layers. The general architecture of the model can be detailed in Figure 1 . In the shared layers, *X* is converted into an embedding vector, then the attention mechanism (used in Transformers) is applied to obtain context information. This information is encoded in a vector for each token $(h_0, ..., h_n)$.

The linear task-specific layers are on top of the shared layers. The fine-tuning process of the shared and task-specific layers is performed using multiple training objectives.

2.1. Training objectives

2.1.1 Sentence similarity

This task is similar to regression. Given a pair of sentences *X1*, *X2*, the function *sim(X1, X2)* predicts a real-valued score indicating the similarity between the sentences. Consider h_0, which corresponds to the contextual embedding of the token *[CLS]*; this vector can be viewed as the semantic representation of (*X1*, *X2*) [5]. The parameters W_{sim}, b are introduced using a fully connected layer to compute the similarity value.

$$sim(X1, X2) = W_{sim} h_0 + b$$

The loss function for this task is the mean squared error (MSE) between the similarity score $sim(X1, X2)$ and the ground-truth value.

2.1.2 Relation extraction

This task aims to predict relations and their types between two entities present in the sentences. This task can be viewed as sentence classification by replacing the entities with predefined tags (e.g. *@CHEMICAL$*, *@GENE$*). For example, the sentence "Citalopram against the RT1-76-induced inhibition of SERT binding" would be replaced

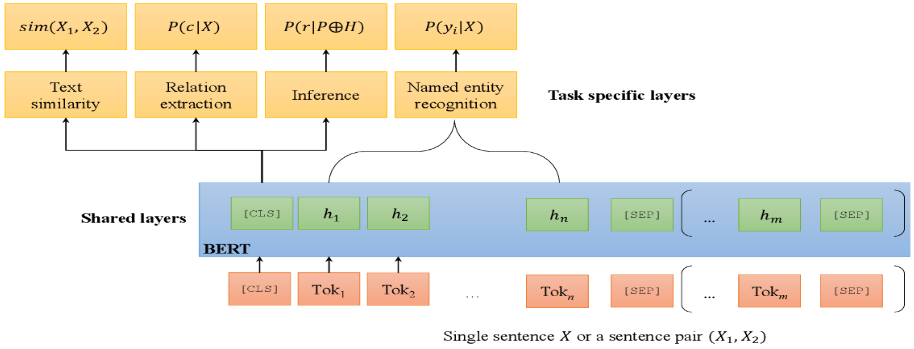

Figure 1. MTL BERT-based model arquitecture

with "@*CHEMICAL$* against the RTI-76-induced inhibition of @*GENE$* binding." In this sentence, there is a chemical-gene relation X with respect to "Citalopram" and "SERT" [7].

Suppose h_0 is obtained from BERT layers, then the probability that a relation X is labeled as class C is obtained with a linear layer (with parameters W_{rel}, b) and the softmax function.

$$P(c|X) = softmax(W_{rel}h_0 + b)$$

The loss function for this task is the category cross-entropy loss, defined as:

$$Loss = -\sum_c \delta(y_c = \tilde{y}) log(P(c|X))$$

where $\delta(y_c = \tilde{y}) = 1$ if the classification $\tilde{y}$ of X is the correct ground-truth of class c, otherwise is 0.

2.1.3 Inference task

The goal of inference is to predict whether the premise sentence entails or contradicts the hypothesis sentence [7]. In this task, X corresponds to the union between the premise and the hypothesis. Inference involves finding a logical relation R between the premise and hypothesis, in other words, $P(R|X)$ where $X = P \oplus H$. Both $P(R|X)$ and the loss function are obtained in the same way as relation extraction.

2.1.4 Named entity recognition

The objective of this task is to predict mention spans in a text. It is similar to machine translation, which predicts tags for each token in a sequence [3]. Given the output embeddings of the shared layers $[h_i]_{i=1}^N$, the model predicts the sequence of labels using the softmax function and the output of the fully connected layer.

$$P(\tilde{y}_{i=j}|X) = \frac{exp(h_i W_j)}{\sum_{i=1}^{L} exp(h_i W_i)}$$

where L is the total number of tags. The loss function corresponds to the categorical cross-entropy loss.

2.2. Model training

To train the model, we first initialize the parameters of the shared layers using the pre-trained BlueBERT model. Subsequently, all layers are refined using Multi-Task Learning. These steps are detailed in the following algorithm:

Algorithm 1 Model initialization and Multi-Task Learning

```
1) Load the pre-trained BERT model to initialize the shared layer parameters.
2) Initialize the task layers parameters randomly
3) Merge mini-batches ($b_t$) from all datasets into a single set $D$.
for each epoch do
    Shuffle $D$
    Calculate $Loss(w)$ given the task of the batch $b_t$
    Backpropagate to obtain $\nabla w$
    $w \leftarrow w - \eta \nabla w$            ▷ $\eta$ : learning rate
end for
```

Finally, we proceed with the fine-tuning of the model obtained in the previous step by continuously training all the layers for each task.

The models evaluated in the experiments are the following: 1) *BlueBERT*: base model directly fine-tuned on each task benchmark. 2) *MT-BlueBERT-Refinement*: Model trained using Algorithm 1 (initialized with BlueBERT). 3) *MT-BERT-Fine-Tuned*: Model trained using Algorithm 1 (initialized with BlueBERT) plus additional fine-tuning on each task benchmark. 4) *MT-BioBERT-Fine-Tuned*: Model trained using Algorithm 1 (initialized with BioBERT) plus additional fine-tuning on each task benchmark.

Models with the *clinical* subindex refer to models that were pretrained in PubMed abstracts and MIMIC-III clinical notes. On the other hand, models with *biomedical* subindex refer to models that were pretrained in biomedical texts.

Table 1. Evaluation results on all benchmarks

Model	BlueBERT biomedical	BlueBERT clinical	MT-BioBERT-Fine-tuned	MT-BlueBERT-Fine-tuned clinical	MT-BlueBERT-Fine-tuned biomedical
ClinicalSTS	0.845	**0.848**	0.807	0.820	0.807
i2b2 2010	0.744	**0.764**	0.740	0.738	0.748
MedNLI	0.822	0.840	0.831	0.814	**0.842**
ShARe/CLEFE	0.754	0.771	0.812	0.814	**0.830**
ChemProt	0.725	0.692	**0.735**	0.724	0.686
DDI	0.739	0.760	**0.810**	0.808	0.779
BC5CDR disease	**0.866**	0.854	0.849	0.853	0.848
BC5CDR chemical	**0.935**	0.924	0.928	0.928	0.914
Avg.	0.804	0.807	**0.814**	0.812	0.807

3. Data

The authors compared the performance of the MT models on 8 datasets in BLUE benchmark. The datasets can be detailed in Table 2.

Table 2. Datasets from BLUE benchmark

Corpus	Task	Domain	Metric	Size
ClinicalSTS	Sentence similarity	Clinical	Pearson	1,068
ShARe/CLEFE	NER	Clinical	F1	10,898
i2b2 2010	Relation extraction	Clinical	F1	9,414
MedNLI	Inference	Clinical	AUC	14,049
BC5CDR disease	NER	Biomedical	F1	12,850
BC5CDR chemical	NER	Biomedical	F1	15,935
DDI	Relation extracion	Biomedical	F1	4,920
ChemProt	Relation extracion	Biomedical	F1	10,025

4. Results & Discussion

Table 3. Evaluation results on clinical benchmarks

Model	BlueBERT *clinical*	MT-BlueBERT-Refinement *clinical*	MT-BlueBERT-Fine-tuned *clinical*
ClinicalSTS	**0.848**	0.822	0.840
i2b2 2010	**0.764**	0.745	0.760
MedNLI	0.840	0.835	**0.846**
ShARe/CLEFE	0.771	0.826	**0.831**
Avg.	0.806	0.807	**0.819**

Table 4. Evaluation results on biomedical benchmarks

Model	BlueBERT *biomedical*	MT-BlueBERT-Refinement *biomedical*	MT-BlueBERT-Fine-tuned *biomedical*
BC5CDRd	**0.866**	0.824	0.865
BC5CDRc	**0.935**	0.930	0.931
DDI	0.739	0.792	**0.820**
ChemProt	0.725	0.714	**0.729**
Avg.	0.816	0.815	**0.836**

Tables 3 and 4 show that multi-task refinement models have similar results to the base models on average in biomedical and clinical domains. Also, as expected, the results improve when fine-tuning is applied after multi-task refinement.

As observed in Table 1, ShARe/CLEFE and MedNLI datasets benefit from multi-task learning when the models are pre-trained using clinical notes. For this reason, MT-BlueBERT-Fine-Tuned$_{clinical}$ created new state-of-the-art results on these benchmarks. ShARe/CLEFE is likely to benefit from MTL because of the nature of the Named Entity Recognition task. BERT models usually require large amounts of data to obtain acceptable results in this task. MedNLI also benefits from ClinicalSTS by being similar and having the same input sequence.

In the biomedical domain, the results in ChemProt and DDI increase when using multi-task models. The reason is that these benchmarks have the most labels and require large amounts of training data. Consequently, the MT-BioBERT-Fine-tuned model created new state-of-the-art results for these tasks.

However, some tasks do not benefit from multi-task learning: ClinicalSTS and i2b2 2010 in the clinical domain and, BC5CDR chemistry & disease in the biomedical domain. The latter are disadvantaged because they have a large enough dataset to fit the model; thus, more training data increases noise.

On the other hand, models pretrained on clinical notes perform better on clinical benchmarks. This trend is similar in biomedical tasks. Therefore, it is convenient to train separate DNNs on different text genres depending on their application.

Finally, as expected, multi-task models perform better on average than the baseline models. These models may be useful for researchers who encounter difficulties in choosing a suitable model or face a problem with limited training data.

5. Conclusion

Based on the results obtained in Tables 3 and 4, the authors recommend MT models trained with the refinement algorithm and then fine-tuned in the task-specific dataset. Nevertheless, as can be observed from the results, one limitation is that Multi-Task Learning is not beneficial for all tasks in clinical and biomedical domains. Changpinyo et al.

postulate that it is not clear whether the characteristics of the data help predict the effectiveness of MT models. However, the results suggest that MT models could improve by using selected examples of some of the benchmarks.

On the other hand, this paper does not study approaches such as fine-tuning only on specific task layers (decoders), soft parameter sharing and knowledge distillation, methods that may be interesting for future work.

References

[1] M. Crawshaw. Multi-task learning with deep neural networks: A survey. *CoRR*, abs/2009.09796, 2020.

[2] M. R. Khan, M. Ziyadi, and M. Abdelhady. Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. *CoRR*, abs/2001.08904, 2020.

[3] J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449, 2018.

[4] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016, 05 2016. baw068.

[5] X. Liu, P. He, W. Chen, and J. Gao. Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504, 2019.

[6] Y. Peng, Q. Chen, and Z. Lu. An empirical study of multi-task learning on BERT for biomedical text mining. *CoRR*, abs/2005.02799, 2020.

[7] Y. Peng, S. Yan, and Z. Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[8] S. Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[9] X. Wang, Y. Zhang, X. Ren, Y. Zhang, M. Zitnik, J. Shang, C. Langlotz, and J. Han. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752, 10 2018.

[10] K. Xue, Y. Zhou, Z. Ma, T. Ruan, H. Zhang, and P. He. Fine-tuning bert for joint entity and relation extraction in chinese medical text. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 892–897, 2019.

[11] Y. Zhang and Q. Yang. A survey on multi-task learning. *CoRR*, abs/1707.08114, 2017.

Literature Review: An Empirical Study of Multi-Task Learning in BERT for Biomedical Text Mining

Mateo Santiago Rueda Molano

Abstract

IDC coding consists of assigning diagnosis and procedure codes to a set of clinical notes. This task is difficult for current machine learning models because the available datasets show a significant imbalance in the number of clinical notes for each code. Furthermore, state-of-the-art pretrained language models such as BERT generate many out-of-vocabulary (OOV) words and only allow a limited sequence length. The authors proposed the TransICD model to overcome these problems. This model is based on a transformer encoder and pretrained using Word2Vec CBOW. Additionally, Biwas et al. used the LDAM loss function to cope with the data imbalance. Compared to literature models, the proposed model has remarkable improvements in macro-AUC, micro-AUC, and micro-F1 metrics.

1. Introduction

The International Classification of Diseases (ICD) is a health classification system based on alphanumeric codes developed by WHO [10, 14]. ICD is a valuable tool for government agencies and healthcare providers worldwide to monitor the population's health and analyze the functioning and performance of the health care system [5]. Insurance companies also widely use this classification system for funding justifications and insurance claim policies [12].

Currently, physicians and nurses perform ICD coding of clinical notes manually, making it error-prone and time-consuming. The American Health Information Management (AHIMA) reported that the manual ICD coding workflow is expensive and inefficient, and it is imperative to develop automated methods to perform this task [1].

ICD classification task consists of automatically assigning diagnosis and procedure codes for a set of discharge summaries. Since each summary can have several codes, this task is treated as a multi-class classification problem. The main difficulty of this task is a large class (code) imbalance of the commonly used datasets (like MIMIC-III) [9].

Multiple approaches using CNN and RNN have been proposed to address this problem [3, 7, 8, 13]. However, these architectures present difficulty in capturing long-term dependencies. Additionally, recent language models such as BERT have a limitation with sequence length [6] and experience many out-of-vocabulary (OOV) words when representing clinical vocabulary [2].

For this reason, the authors proposed the TransICD model which is based on a transformer encoder with a pretrained Continuous Bag of Words (CBOW) [11]. This model mitigates the problem of OOV words and short sequence length. Moreover, the authors applied the label distribution aware margin (LDAM) [4] loss function to reduce the impact of the dataset code imbalance.

2. Methods

2.1. Model

The model input is a clinical note represented by a vector $W = \{W_i\}_{i=1}^{N}$, where W_i is the vocabulary index of the i-th word, and N is the maximum length of the sequence. This vector W is mapped to an embedding matrix $E \in \mathbb{R}^{N \times D_H}$, where D_H is the embedding dimension.

The multi-headed self-attention mechanism is applied to matrix E through a transformer encoder. The output H of the encoder corresponds to the contextual representation of the word. To transform this representation into a multi-label classification problem, self-attention is applied to matrix H. For this purpose, the attention weights A are calculated with the following equation:

$$A = softmax(tanh(HU)V)$$

where $U \in \mathbb{R}^{D_H \times D_A}$ and $V \in \mathbb{R}^{D_A}$ are model parameters, and D_A is a hyper-parameter.

The code-specific representation $C \in \mathbb{R}^{L \times D_H}$ of input W is obtained by multiplying weight matrix A with H.

$$C = H^T A$$

Additionally, the authors employed the LDAM approach to cope with the dataset code imbalance. In this approach, the predictions $\hat{y}$ are obtained using a linear layer summed with an additional term, followed by the sigmoid function.

$$\hat{y}_l = \sigma(ZC_l + b - \mathbf{1}(y_l = 1)\nabla l)))$$

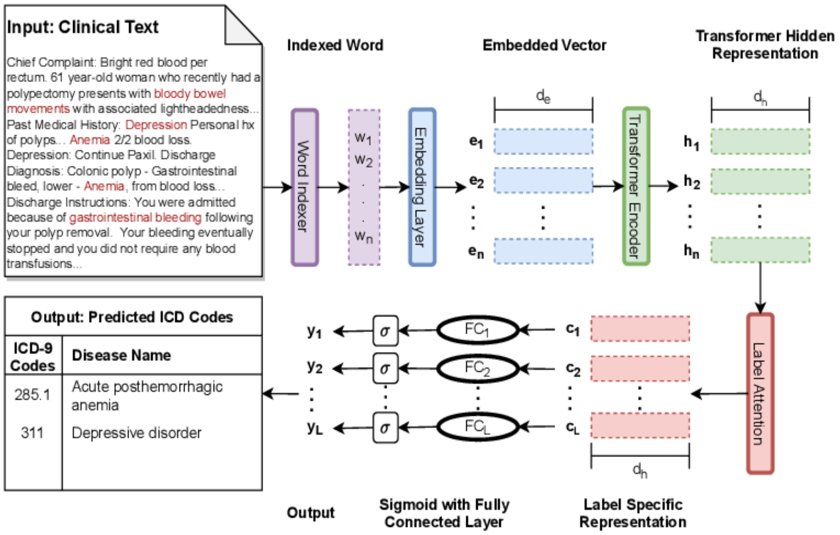
Input: Clinical Text
Chief Complaint: Bright red blood per rectum. 61 year-old woman who recently had a polypectomy presents with bloody bowel movements with associated lightheadedness... Past Medical History: Depression Personal hx of polyps... Anemia 2/2 blood loss. Depression: Continue Paxil. Discharge Diagnosis: Colonic polyp - Gastrointestinal bleed, lower - Anemia, from blood loss... Discharge Instructions: You were admitted because of gastrointestinal bleeding following your polyp removal. Your bleeding eventually stopped and you did not require any blood transfusions...
Indexed Word
Embedded Vector
Transformer Hidden Representation
Word Indexer
w_1
w_2
w_n
Embedding Layer
d_e
e_1
e_2
e_n
Transformer Encoder
d_h
h_1
h_2
h_n
Label Attention
Output: Predicted ICD Codes
ICD-9 Codes
Disease Name
285.1
Acute posthemorrhagic anemia
311
Depressive disorder
y_1
y_2
y_L
σ
FC_1
FC_2
FC_L
c_1
c_2
c_L
d_h
Output
Sigmoid with Fully Connected Layer
Label Specific Representation

Figure 1. TransICD model arquitecture

$$\nabla l = \frac{d}{n_l^{1/4}}$$

In this equation, Z,b are the linear layer weights, $\mathbf{1(.)}$ function outputs 1 if $y_l = 1$, d is a constant term and n_l is the counter of inputs having the label l. The loss function corresponds to multi-label cross-entropy loss between the predictions obtained in the previous step ($\hat{y}$) and the true labels (y). The graphical representation of the model can be detailed in Figure 1.

The metrics to evaluate the models were micro-averaged and macro-averaged area under the ROC curve (AUC) and F1 score. The difference is that micro-averaged is computed considering each pair (document, label) as a single prediction, while macro-averaged is computed using the metric mean for each label. Additionally, the authors included precision at k=5 which computes the percentage of true labels in the top-5 predictions.

3. Data

MIMIC-III is a dataset composed of many publicly available medical records. Each record includes ICD-9 codes of the procedures and diagnoses performed on the patient. This dataset has been widely used for ICD multi-label classification.

Due to the imbalance of codes in this dataset, the authors only considered MIMIC-III 50 settings, containing only the 50 most frequent ICD codes. The preprocessing of the datasets comprised the following operations: lowercasing, punctuation removal, stopword elimination, and stemming. Subsequently, the authors used Word2Vec CBOW to obtain the Word embeddings (size 128) by training the entire set of clinical notes. The dataset was divided into training, validation, and evaluation with 8066, 1573, and 1729 examples, respectively.

4. Results & Discussion

Table 1. Evaluation results (in %) on the MIMIC-III dataset

Model	AUC		F1		Precision
	Micro	Macro	Micro	Macro	k =5
Bi-GRU	82.8	86.8	48.4	54.9	59.1
C-MemNN	83.3	-	-	-	42.0
C-LSTM-Att	-	90.0	-	53.2	-
CNN	87.6	90.7	**57.6**	62.5	**62.0**
CAML	87.5	90.9	53.2	61.4	60.9
LEAM	88.1	91.2	54.0	61.9	61.2
Transformer	85.2	88.9	47.8	56.3	56.5
Transformer + Label Attention	88.2	91.1	49.4	59.3	59.6
Transformer + Label Attention + LDAM	**89.4** ± 0.1	**92.3** ± 0.1	56.2 ± 0.4	**64.4**± 0.3	61.7 ± 0.3

The table 1 shows the results of the TransICD model compared to other literature models for this problem. The TransICD results correspond to the mean and standard deviation of 5 runs of the model, initializing the parameters with different random seeds. The low standard deviation shows that the model is consistent to fit the data.

TransICD obtained the best results in the macro-AUC, micro-AUC, and micro-F1 metrics, while the results in F1-macro and precision at K=5 are slightly below the best result (CNN approach). The low results on the macro-F1 metric among all Transformer models suggest that these models have difficulty predicting infrequent codes. However, these results are also low across all baselines, which is evidence of the task's difficulty.

As shown in table 1, the baseline Transformer model has more favorable results than Bi-GRU on the macro-AUC, micro-AUC, and micro-F1 metrics. The reason is probably

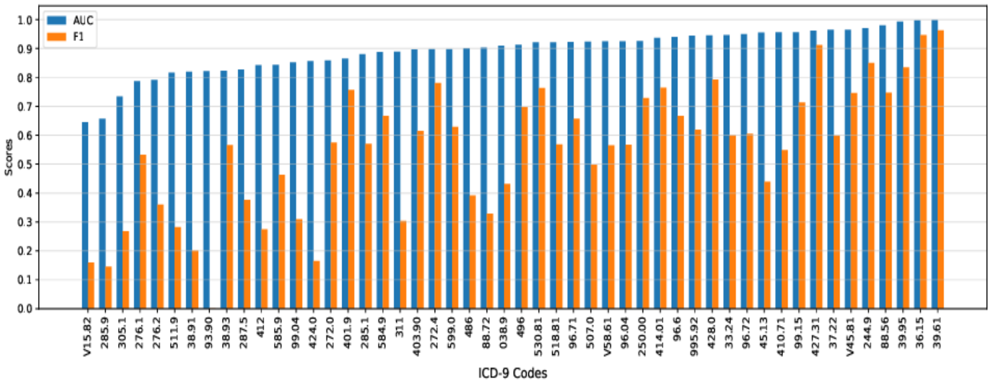

Figure 2. AUC and F1 scores across all codes in the MIMIC-III dataset.

that the Transformer can better capture long-term dependencies rather than recurrent units.

The results show that both model modifications (Label attention and LDAM) improve the performance of the baseline Transformer model. The substantial increase of all metrics between Transformer + Attention and TransICD models proves that LDAM is very important to cope with the dataset's long-tail distribution of codes.

As shown in Figure 2, TransICD has favorable results in the AUC metric for most of the codes. For 90% of the codes, the model obtains an AUC greater than 0.8. Only 4% of the codes have an AUC less than 0.7. However, the results are not promising in the F1 metric, with only 10% of the codes having an F1 greater than 80. The authors found that several of the codes with negative results have a low frequency or are codes that are usually difficult to classify by medical personnel. For example, the codes *Tobacco use disorder* and *History of tobacco* are frequently mislabeled.

5. Conclusion

This paper presents a method to automatically predict the ICD codes on the MIMIC-III dataset. The authors propose to create the word embeddings using Word2Vec CBOW in response to the limited sequence length and the large number of out-of-vocabulary words when using pre-trained BERT models. The model trained with LDAM loss has more favorable results on all metrics, demonstrating this function's benefit for training models on unbalanced datasets. Finally, the label attention mechanism shows positive results in the multi-label prediction task. However, the results on this dataset are not promising for infrequent codes so future work should focus on models that can fit the data using few-shot learning.

References

[1] AHIMA. Delving into computer-assisted coding, 2013.

[2] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, and M. McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[3] W. Bao, H. Lin, Y. Zhang, J. Wang, and S. Zhang. Medical code prediction via capsule networks and icd knowledge. *BMC Medical Informatics and Decision Making*, 21(S2), 2021.

[4] K. Cao, C. Wei, A. Gaidon, N. Aréchiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *CoRR*, abs/1906.07413, 2019.

[5] CDC. Icd-10-cm official coding guidelines - supplement coding encounters related to covid-19 coronavirus outbreak, 2020.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] J.-L. Hsu, T.-J. Hsu, C.-H. Hsieh, and A. Singaravelan. Applying convolutional neural networks to predict the icd-9 codes of medical records. *Sensors*, 20(24), 2020.

[8] S. Hu and F. Teng. An explainable CNN approach for medical codes prediction from clinical text. *CoRR*, abs/2101.11430, 2021.

[9] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, and et al. Mimic-iii, a freely accessible critical care database, 2016.

[10] R. Kaur, J. A. Ginige, and O. Obst. A systematic literature review of automated ICD coding and classification systems using discharge summaries. *CoRR*, abs/2107.10652, 2021.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[12] K. R and G. JA. Comparative analysis of algorithmic approaches for auto-coding with icd-10-am and achi, 2018.

[13] S.-M. Wang, Y. hsuan Chang, L.-C. Kuo, F. Lai, Y.-N. V. Chen, F. yun Yu, C.-W. Chen, Z. wei Li, and Y.-F. Chung. Using deep learning for automatic icd-10 classification from free-text data. 2020.

[14] C. Yan, X. Fu, X. Liu, Y. Zhang, Y. Gao, J. Wu, and Q. Li. A survey of automated icd coding: Development, challenges, and applications. *Intelligent Medicine*, 2022.