Journal of Electronic Research and Application, 2024, Volume 8, Issue 3
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Adapter Based on Pre-Trained Language Models for Classification of Medical Text

Quan Li*

University of Science and Technology of China, Hefei 230026, Anhui Province, China

***Corresponding author:** Quan Li, SA21011062@mail.ustc.edu.cn

Abstract: We present an approach to classify medical text at a sentence level automatically. Given the inherent complexity of medical text classification, we employ adapters based on pre-trained language models to extract information from medical text, facilitating more accurate classification while minimizing the number of trainable parameters. Extensive experiments conducted on various datasets demonstrate the effectiveness of our approach.

Keywords: Classification of medical text; Adapter; Pre-trained language model

Online publication: June 14, 2024

1. Introduction

Text classification, a fundamental task in natural language processing, swiftly categorizes vast data into relevant categories, enabling automatic classification [1]. Thus, it plays a pivotal role in text data retrieval and mining. However, in specialized domains such as the medical field, text classification may encounter data sparsity issues. Medical text encompasses a variety of documents, including medical records and literature [2]. Medical records document the entire medical process, including a doctor's examinations, diagnoses, treatments, and the progression of a patient's disease. They provide detailed information about a patient's medical history and the efficacy of prescribed treatments, serving as crucial resources during treatment. On the other hand, medical literature consists of research findings regarding the latest medical methods. Additionally, medical literature comprises research findings on the latest medical methods. Medical text typically incorporates normalized medical terminology, including concepts or abbreviations specific to the medical field, such as "blood pressure of 120/55." Moreover, medical records often feature sentences with poor grammatical structure [3,4]. Consequently, text classification within the medical domain presents unique challenges.

Deep learning methods have demonstrated remarkable performance in tasks such as image classification and speech recognition, leading to their widespread adoption in natural language processing (NLP) in recent years, yielding significant improvements. Pretrained language models (PLMs) have particularly revolutionized natural language understanding [5]. Unlike traditional neural networks, one key distinction of PLMs is their capacity to comprehend and generate natural language [5].

We employed a PLM-based approach to categorize text fragments at the sentence level, leveraging emergent semantics extracted from a corpus of medical text. Recognizing the substantial training cost associated with fine-tuning PLMs, we proposed an adapter based on PLMs for classifying medical text, featuring a small set of trainable parameters. This method harnessed the capabilities of PLMs to address medical text classification while mitigating training costs by solely updating the adapter parameters and freezing all PLM parameters. Experiments on real data were conducted to evaluate efficiency and effectiveness.

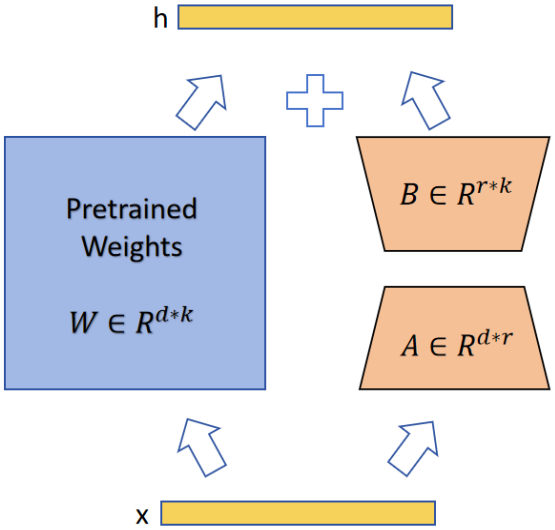

Figure 1. In an adapter-based model, we only train A and B.

2. Related works

2.1. Pre-trained language models

The recent breakthrough in self-supervised pre-trained language models has boosted the development of natural language processing. Decoder-based language model such as GPT-2 [6] leverages the Transformers architecture to pretraining on large-scale web texts. Encoder-based language model such as BERT [7] proposes masked language modeling and create the pre-training/fine-tuning paradigm. Training larger language models generally results in better performance and remains an active research direction.

2.2. Adapter structure in deep learning

Some scholars proposed inserting adapter layers between existing layers in a neural network [8-10]. These methods reduce memory requirements by using a small set of trainable parameters, while not updating the full model parameters which remain fixed. Since the memory overhead of the adapter is minimal, some recent methods use more adapters to improve performance without significantly increasing the total memory used [11].

3. Preliminaries

We analyze models pre-trained on masked language modeling (MLM) objectives. Let U denote a finite vocabulary of input tokens, $X = (X_1, ..., X_1) \in U^*$ represent the set of variable-length sequences of tokens, and be a random sequence of T tokens. Let $\Delta^{|U|}$ denote the space of probability distributions over the tokens.

3.1. Pre-training and downstream task

Consider the multi-layer model (MLM) denoted as $G(x)=(G_1(x), G_2(x),...)$, which is responsible for predicting

a probability vector associated with input data x. Additionally, let ($G(x^1)$, $G(x^2)$,...) represent the MLM that predicts probability vectors for the input sequence x^i. Each component G_i calculates the distribution of the i-th token, denoted as X_i, conditioned on all other tokens: $G_i(x)=P[X_i|X_{-i}=x_{-i}]$ Here, $P[X_i|X_{-i}=x_{-i}]\in\Delta^{|U|}$ signifies a probability vector. The main task at hand involves labeled examples in the form of pairs (x, $H^*(x)$). Here, $H^*:X->Y$ serves as the source of ground-truth labels for downstream tasks, and Y represents a discrete set of classification labels.

3.2. Lora-based adapter for pre-trained language model

A neural network contains many dense layers which perform matrix multiplication. The weight matrices in these layers typically have full rank. For a pre-trained language model's weight matrix $W_0\in R^{d*k}$, we can train its update by representing the latter with a low-rank decomposition $W_0+\Delta W=W_0+BA$, where $B\in R^{d*k}$, $A\in R^{d*k}$, and the rank $r<<min(d,k)$. During training, W_0 is frozen and does not receive gradient updates (**Figure 1**), while A and B contain trainable parameters. Note both W_0 and $\Delta W=BA$ are multiplied with the same input, and their respective output vectors are summed coordinate-wise. For $h=W_0x$, our modified forward pass yields:

$$h=W_0x+\Delta Wx=W_0x+BAx. \quad (1)$$

We used a random Gaussian initialization for A and zero for B, so $\Delta W=BA$ is zero at the beginning of training.

3.3. Other notations

Let Δ^d denote the space of d-dimensional probability vectors. We use $P[V]\in\Delta^{|V|}$ to denote the distribution of V, and $P[U|V=v]\in R^{|U|}$ to signify the conditional distribution of U given $V = v$,$Pr(V=v)\in[0,1]$ denotes the probability that V assumes the specific value v. For a sequence $v=(v^1,...,v^t)$, we use the notation $v^{(i:j)}$ for $i \leq j$ to denote $(v^i,...,v^j)$, and v_{-i} to denote $(v^{(1:i-1)},...,v^{(i+1:t)})$.

4. Methodology

4.1. Problem statement

In this section, we formulated the classification problem of medical text-based large language model. Consider a set of input sentences containing T tokens, denoted as $X_{(1:T)}$. The classification problem can be formulated as a sequence labeling task that assigns a label to the sentence.

Suppose we have a sequence with T tokens, $X_{(1:T)}$. The classification problem aims to find the underlying true label y with a set of trainable parameters of large language model, θ.

4.2. Model architecture

Given a pre-trained language model M, a sequence of discrete input tokens $x_{(1:n)}=<x_0, x_{1,...,}x_n>$ will be mapped to $<e(x_0), e(x_1),..., e(x_n)>$ by the pre-trained language model . In the classification scenario, condition on the context x, we often use the output embeddings of a set of target tokens y for classification processing. For instance, in the pre-training, x refers to the unmasked tokens while y often refers to the [MASK] ones; and in the classification,x refers to the sentence tokens while y often refers to the [CLS] (**Figure 2**).

Let U be the vocabulary of a language model M. For simplicity, given a template $T=<x,y>$. we can optimize the trainable parameters of M with the classification loss function L by

$$\theta = argmin\ L(M(x,y)) \quad (2)$$

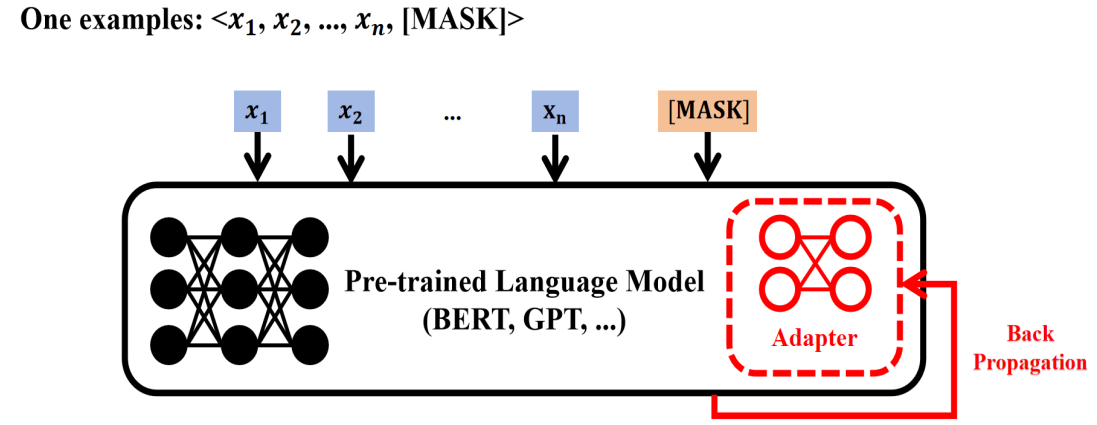

Figure 2. An example of a classification problem-based pre-trained language model with a trainable adapter. The model consists of two components: a pre-trained model and an adapter. The parameters of the pre-trained model were frozen and the parameters of the adapter were trainable.

In the selection of the pre-trained model, we employed the two most popular transformer-based pre-trained models as the base models, including an encoder-based model (BERT) and a decoder-based model (GPT-2). For the selection of adapter, we employed the most popular adapter methods, LoRA [9] and AdaLoRA [10]. So, the whole model consisted of a pre-trained language model and an adapter.

4.3. Model training

Our model consisted of a pre-trained language model and an adapter. We performed supervised training using the available labeled data *y*. We employed an integrated model to generate the flag , allowing us to see the whole model within a small set of trainable parameters. In other words, we calculated the Loss function based on the discrete set *Y*, in which estimated *y* takes value. The Loss function is as follows:

$$L = -\frac{1}{N}\sum_{y'} yln(y') + (1-y)ln(1-y') \quad (3)$$

Where the y' represents the prediction by pre-trained language with an insert adapter and the y is the truth flag.

When updating the whole model, we only updated the trainable parameters of the whole model. We froze all parameters in the pre-trained language model and set the parameters in the adapter model as trainable parameters.

5. Results

5.1. Experiment settings

(1) Datasets

We used a real dataset comprising 6000 training data entries of medical text and 1000 testing data entries of medical text. We needed to extract key information from a massive amount of literature for disease diagnosis and treatment recommendations and determine whether a piece of text belongs to the medical domain.

Pre-trained language models. We select two different architecture models to be the pre-trained language models, including an encoder-based model (BERT)[7] and a decoder-based model (GPT-2) [6]. We inserted an adapter called “Lora” into the pre-trained language model to finish the classification of

medical text.

(2) Implementation

Algorithms were implemented in Python and run on a PC with an Intel(R) Xeon(R) CPU E5-2698v4 @ 2.20GHz, an NVIDIA A100, and 256GB main memory. During the model training stage, we took learning rates from 1e-5, 2e-5, and 3e-5 and batch sizes from 16, 32, 64, and 128. We fine-tuned the whole model from 20 epochs within a small set of trainable parameters. We evaluated the performance of every epoch. We used early stopping to avoid overfitting.

5.2. Evaluation across models

(1) BERT base/large

BERT has found applications in some natural language processing tasks. While BERT has been surpassed by much larger models on NLP leaderboards, such as the GLUE benchmark, in recent years, it remains a competitive and popular pre-trained model among practitioners due to its manageable size. We utilize the pre-trained BERT base (110M) and BERT large (340M) models from the HuggingFace Transformers library to classify medical texts. A sequence length of 512 is employed. The results of BERT-LoRA and BERT-AdaLoRA can be found in **Table 1**.

(2) GPT-2 medium/large

Having demonstrated the good performance of the encoder-based model, we used the decoder-based model with the LoRA adapter to classify the medical texts We obtained the pre-trained GPT-2 medium (355M) and GPT-2 large (774M) models from the HuggingFace Transformers library. We utilize a sequence length of 512. The results of GPT2-LoRA and GPT2-AdaLoRA can be found in **Table 2**.

Table 1. Precision of BERT base/large with LoRA/ AdaLoRA adapters

Model & method	# Trainable parameters	Precision
BERT base (LoRA)	0.6M	97.7%
BERT base (AdaLoRA)	0.4M	96.1%
BERT large (LoRA)	1.6M	98.9%
BERT large (AdaLoRA)	1.2M	96.7%

Table 2. Precision of GPT-2 medium (M)/large (L) with LoRA/ AdaLoRA adapters

Model & method	# Trainable parameters	Precision
GPT-2 M (LoRA)	1.6M	98.6%
GPT-2 M (AdaLoRA)	1.2M	97.8%
GPT-2 L (LoRA)	3.0M	99.8%
GPT-2 L (AdaLoRA)	2.2M	98.7%

6. Conclusion

In this paper, we tackled the classification of medical text, leveraging the capabilities of pre-trained language models to address challenges within the medical domain. To address practical constraints such as the limited number of trainable parameters in the model, we proposed a novel framework for training pre-trained language models. Our aim was to train these models to handle medical domain problems using only a small set of

trainable parameters. We introduced an adapter method, which involved inserting trainable layers into the pre-trained language model to simulate the updating process of the entire pre-trained language model within a limited set of trainable parameters. Extensive experiments conducted on real datasets demonstrated the effectiveness of our approach.

Disclosure statement

The author declares no conflict of interest.

References

[1] Alishahi A, Chrupała G, Linzen T, 2019, Analyzing and Interpreting Neural Networks for NLP: A Report on the First BlackboxNLP Workshop. Natural Language Engineering, 25(4): 543–557.

[2] Mujtaba G, Shuib L, Idris N, et al., 2019, Clinical Text Classification Research Trends: Systematic Literature Review and Open Issues. Expert Systems with Applications, 116: 494–520.

[3] Kaurova O, Alexandrov M, Blanco X, 2011, Classification of Free Text Clinical Narratives (Short Review). Business and Engineering Applications of Intelligent and Information Systems, 2011: 124–135.

[4] Hoang N, Patrick J, 2016, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 13–17, 2016: Text Mining in Clinical Domain: Dealing with Noise. San Francisco, 549–558.

[5] Brown TB, Mann B, Ryder N, et al., 2020, Language Models are Few-Shot Learners. arXiv. https://doi.org/10.48550/arXiv.2005.14165

[6] Radford A, Wu J, Child R, et al., 2019, Language Models are Unsupervised Multitask Learners. OpenAI blog, 1(8): 9.

[7] Devlin J, Chang M-W, Lee K, et al., 2019, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 2–7, 2019: BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. Minneapolis, 4171–4186. Association for Computational Linguistics.

[8] Houlsby N, Giurgiu A, Jastrzebski S, et al., 2019, Parameter-Efficient Transfer Learning for NLP. International Conference on Machine Learning, 2790–2799.

[9] Hu E J, Shen Y, Wallis P, et al., 2021, Lora: Low-Rank Adaptation of Large Language Models. arXiv. https://doi.org/10.48550/arXiv.2106.09685

[10] Zhang Q, Chen M, Bukharin A, et al., 2022, AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. arXiv. https://doi.org/10.48550/arXiv.2303.10512

[11] Dettmers T, Pagnoni A, Holtzman A, et al., 2024, QLoRA: Efficient Finetuning of Quantized LLMs. arXiv. https://doi.org/10.48550/arXiv.2305.14314