Using Parameter Efficient Fine-Tuning on Legal Artificial Intelligence

Kuo-Chun Chien[1], Chia-Hui Chang[1] and Ren-Der Sun[1]

[1]National Central University, No. 300, Zhongda Rd., Zhongli District, Taoyuan City 320317, Taiwan (R.O.C.)

Abstract

Legal AI has a wide range of applications such as predicting whether a prosecution will be punished, or whether the punishment will be a prison sentence or a fine. However, current advances in natural language processing have resulted in an ever-increasing number of language models. The cost of fine-tuning the pre-trained language model and storing these fine-tuned language models becomes more and more expensive. To address this issue, we adopted the concept of Parameter Efficient Fine-Tuning (PEFT) and applied it to the field of Legal AI. By leveraging PEFT techniques, particularly through the implementation of the Low-Rank Adaptation (LoRA) architecture, we have achieved promising results in fine-tuning pre-trained language models. This approach enables us to achieve comparable, if not superior, performance while significantly reducing the time required for model adjustments. It demonstrates the potential of PEFT techniques in adapting language models to different legal frameworks, enhancing the accuracy and relevance of legal knowledge services, and making Legal AI more accessible to individuals without legal backgrounds.

Keywords

Legal AI, Legal Judgment Prediction, Parameter-Efficient Fine-Tuning,

1. Introduction

Legal AI refers to the utilization of artificial intelligence (AI) technology in the legal sector. It is an expanding field that harnesses sophisticated algorithms and machine learning techniques to assist in the organization, analysis, and interpretation of extensive legal documentation. Applications of legal AI encompass various areas, including case management [1], legal judgment prediction (LJP) [2, 3], court views generation [4], among others. From overseeing compliance to managing legal risks, from streamlining contract management to conducting due diligence, AI technology can automate and enhance the legal workflow, leading to improved efficiency, accuracy, and convenience for legal professionals. Ultimately, the implementation of legal AI has the potential to revolutionize the legal industry, making legal services more accessible and cost-effective for individuals and businesses alike.

Legal cases typically fall into two main categories: civil law and criminal law. Since gathering facts and evidence for civil cases can be challenging [5], most research efforts in LJP have

2nd Workshop on Knowledge Management and Process Mining for Law, 9th Joint Ontology Workshops (JOWO 2023), co-located with FOIS 2023, 19-20 July, 2023, Sherbrooke, Québec, Canada

qk0614@gmail.com (K. Chien); chiahui@g.ncu.edu.tw (C. Chang); renkensun40@gmail.com (R. Sun)
https://sites.google.com/site/jahuichang/ (C. Chang)
0000-0002-1101-6337 (C. Chang)

primarily concentrated on criminal cases [6, 7, 8], utilizing verdicts as the primary dataset for predicting potential legal articles, charges, and terms based on given factual information. However, in the field of Legal Judgment Prediction (LJP) in criminal cases, there are not only verdicts but also indictments and various prediction tasks. For example, prosecutors may want to know whether the case ultimately went to trial according to the legal provision and charges in the indictment; if the case went to trial, did its punishment result in jail time or a fine; and if the case was dismissed, was it because of immunity or not guilty?

In recent years, significant progress has also been made on many legal tasks based on pre-trained models, including accusation prediction [9], prison term classification [2], criminal element extraction [3], and court view generation [4], etc. However, current advances in natural language processing have resulted in an ever-increasing number of language models. The cost of fine-tuning the pre-trained language model for different LJP tasks and storing these fine-tuned language models becomes more and more expensive. If we were to train a separate large language model for each sub-task, it would consume excessive time and resources. This highlights the need for adaptive methods, such as Parameter-Efficient Fine-Tuning (PEFT), which allows for selective updates or additions of parameters to train the model for new tasks.

In this study, we propose the use of PEFT to fine-tune pre-trained language models. Specifically, we adopt Low-Rank Adaptation of Large Language Models (LoRA)[10], as an implementation of PEFT, which offers advantages in reducing computational resources and fine-tuning time while maintaining or surpassing model performance. This makes it particularly valuable for refining large models with billions of parameters.

The rest of the paper is organized as follows: Section 2 introduces related work on legal AI and LJP and Parameter-Efficient Fine-Tuning (PEFT). The problem definition and dataset construction is detailed in Section 3. Section 4 explains PEFT. We report the experimental results in Section 5. Finally, Section 6 concludes the paper and suggests for future research direction.

2. Related Work

2.1. Legal AI

Legal artificial intelligence (LegalAI) has drawn increasing attention from NLP researchers because of the vast amount of legal documents. Zhong et al. [11] surveyed the researches on legal artificial intelligence (LegalAI) and categorized its applications into three types: legal judgment prediction (LJP), similar case matching, and legal question answering.

Among them, legal judgment prediction has been widely studied for decades, and there are also several related LJP datasets, such as CAIL [2], CAIL-Long[12], ECHR[13, 14], etc. CAIL is the first Chinese Legal Judgment Prediction Dataset, which collects the criminal cases from Supreme People's Court of China. CAIL-Long further obtains more information form Supreme People's Court of China, including civil and criminal cases. ECHR [14] is an English Legal Judgment Prediction Dataset collected from European Court of Human Rights, which contains cases that a state has breached human rights provisions of the European Convention of Human Rights.

LegalAI's research methods can be divided into symbol-based methods and embedding-based methods [11]. In the past, researchers have used traditional machine learning methods for feature extraction, attempting to extract or create specific features from the description of criminal facts using additional labeling to help describe the crime. For example, Hu et al. [6] combined ten discriminative legal features to help predict low-frequency charges. Shaikh et al. [15] identified and extract 19 features of murder-related criminal cases to train a binary classifier to judge if guilty or not. However, these features are difficult to apply to large-scale datasets [16] because fact descriptions are expressed in different ways and some of these features require additional labels.

To address the above scaling issues, researchers have attempted to incorporate legal knowledge into neural networks via automatic learning. For example, Luo et al. [17] adopted a two-step approach to filter out irrelevant law articles with and retain the top k articles to scale up to a large number of law articles. They built a binary classifier for each article focusing on its relevance to the input case. The advantage of such an approach is that we can add new articles with the existing classifiers untouched. Similarly, Bao et al. [9] proposed an attention neural network, LegalAtt, which uses relevant articles to improve the performance and interpretability of charge prediction task. Gan et al. [18] injected the legal knowledge in the form as a set of first-order logic rules and integrate these rules into a co-attention network-based model, which makes the prediction more interpretable for civil loan cases. Kang et al. [16] constructed auxiliary fact representations from the definitions of behavioral reasons to enhance fact descriptions. Lyu et al. [3] introduced four types of criminal elements as bridges between the fact description and article, and used the concept of reinforcement learning to jointly identify similar articles and confusing fact descriptions in the legal judgment prediction task.

Multi-task learning framework is a machine learning method that can train multiple related tasks simultaneously, thus improving the performance of each task. It can use a shared layer to extract common features for all tasks, and then use different specialized layers to handle the details of each task, or use different layers to extract features for each task and then use some methods to limit the differences between the parameters of these layers. Zhong et al. [7] proposed the TopJudge model, which uses a topological graph to enhance performance by exploiting the relationships between legal judgments, predicting articles, charges, and terms. Yang et al. [19] proposed a multi-layer forward prediction and backward validation framework to effectively utilize the dependency relationships between multiple sub-tasks.

2.2. Parameter-Efficient Fine-Tuning

It has been shown that it is feasible to update or add a very small number of parameters as opposed to updating all of the parameters of the pre-trained model as is the case with ordinary fine-tuning. The addition of adapters, which are tiny trainable feed-forward networks inserted between the layers of the fixed pre-trained model, was suggested Houlsby et al. [20] (See Figure 1). Since then, a wide range of advanced PEFT techniques have been put forth, e.g. low rank adaptation by Hu et al. [10], and prefix-tuning by [21]. In a way, Houlsby et al. (2019) places two adapters sequentially within one layer of the transformer, that is to say typical adapters are sequential computation. On the other hand, prefix-tuning and LoRA can be thought of as a "parallel" computation to the PLM layer. An unified view toward Parameter-efficient transfer learning was proposed by He et al. [22].

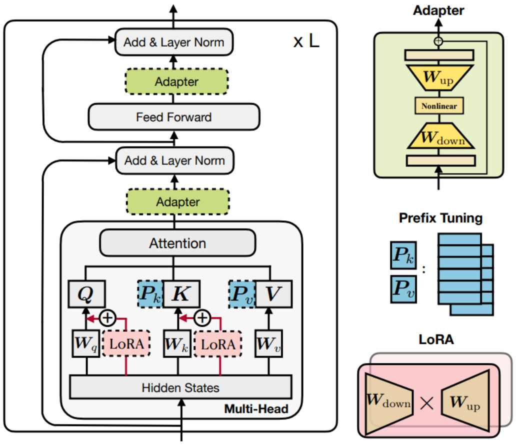

Figure 1: Three PEFT mechanism: Adapter, prefix tuning and LoRA (Adopted from [22])

3. Problem Formulation and Dataset Construction

Four steps make up a criminal proceeding: investigation, prosecution, trial, and execution. Among these steps, the public is most interested in the investigation and trial steps. The investigation procedure refers to the process in which law enforcement agents look into potential criminal events and gather evidence under the direction of the prosecutor. The prosecution will file charges and begin the trial process if they feel that the defendant has a strong suspicion of committing a crime. An impartial, unbiased judge oversees the trial process and determines whether the defendant actually committed a crime based on the evidence given by the prosecutor. Today, judgment documents are used as the data source in the majority of publicly accessible datasets for LJP research. However the language employed in judgment documents is frequently more eloquent, and the substance primarily concentrates on the facts and procedures, leading to greater document lengths and more difficult comprehension for legal specialists. On the other hand, prosecutors employ language that is shorter and more akin to that of the general public when describing the portion of the criminal facts in the indictment that are based on their involvement in the investigation. Hence, rather than using judgment documents for the scope of the data collection, we employ indictments.

3.1. Dataset Construction

We collected indictments from the public document inquiry system of the Ministry of Justice of Taiwan from June 15, 2018 to June 30, 2021. The defendant, charges, criminal facts, and legal provisions were extracted from the indictments using regular expressions, and the material was then organized into a JSON format. There were 533 articles under 41 laws and 183 charges from 355,295 cases in the original dataset.

How many articles and charges to include in the prediction model is a recurring issue while creating the LJP dataset. We screened out instances where the number of charges or articles was insufficient in order to make the experiment fair and prevent classification-related insufficient training or testing data, which may have an impact on the experimental outcomes (e.g., less than 30 cases). Furthermore, the first 100 articles of Taiwan's criminal code contain definitions of terms like attempted offenses and criminal responsibility, but we did not include these articles in our dataset because they do not specify the real penalties. Excluding the above cases, the total number of articles decreased significantly to 165, and the number of charges decreased from 183 to 94. A total of 12,541 cases were removed, accounting for 3.5% of the total dataset. It is worth noting that a case may violate more than one charge, but often only the primary charge are listed in the indictment. Thus, it is more difficult to estimate charge than articles (even though the number of articles in our dataset is greater than the number of charges). The distribution of instances in this dataset is unequal, as one might anticipate. The top 10 counts make about 85% of all cases, according to the number of charges in the indictment. In contrast, just 0.14% of the instances are covered by the lowest 10 charges. We divided the cases into categories based on the charges in the indictment in order to fairly split the data, using 80% of the instances in each category as training data, 10% as validation data, and the final 10% as testing data. Lastly, we created a dataset called TWLJP[1] (TaiWan Legal Judgment Prediction

Facts	…知悉將帳戶存簿、金融卡及密碼交付他人使用，恐為不法者充作詐騙被害人匯入款項之犯罪工具，亦不違背其本意之洗錢及幫助詐欺取財之犯意，將其之存摺及提款卡等資料，並提供提款卡密碼，以寄送包裹之方式，租借寄予詐欺集團成員，容任該人及其所屬之詐騙集團持以犯罪使用。…	…Knowing that handing over account passbooks, financial cards, and passwords to others may serve as tools for criminals to commit fraud by transferring funds, and also not deviating from their intention of money laundering and aiding in fraudulent schemes, providing the passbooks, withdrawal cards, and supplying the PIN codes through parcel delivery to members of a fraudulent group enables that person and their affiliated fraudulent organization to utilize them for criminal purposes. …
Laws	洗錢防制法、刑法	Money Laundering Control Act、Criminal Code
Articles	…是核被告所為，係犯洗錢防制法第 2 條第 2 款、第 14 條第 1 項之洗錢罪嫌及刑法第 30 條第 1 項前段、第 339 條第 1 項之幫助詐欺取財罪嫌…	It was committed by the defendant, and is guilty of the crime of money laundering under Article 2, paragraph 2, and Article 14, paragraph 1, and the crime of assisting in fraudulent acquisition of money under Article 30, paragraph 1, and Article 339, paragraph 1 of the Criminal Code.
Charge	詐欺	Fraud

Table 1
An example indictment document of a criminal case (original Chinese text and its English translation). We have highlighted the criminal intent and the article in blue and green respectively.

[1] doi: 10.17632/gxxcv4jcgg.1

Datasets) by combining the data from all categories to create training, validation, and testing datasets.

Table 1 displays an example of an indictment, with the criminal intent and articles marked in blue and green, respectively. In this instance, a suspect gave the fraudsters access to his bank account, and the group tricked the victim into wiring money to the account before withdrawing it. The Anti-Money Laundering Act and the Criminal Law were both allegedly broken by the defendant, however the indictment only listed fraud as a crime.

Dataset	# cases	# laws	# articles	# charge	avg length	avg articles
TWLJP	342,754	33	165	94	376.31	1.16

Table 2
The Statistic of TWLJP dataset

3.2. Problem Formulation

Let $D = (d_1, d_2, \cdots, d_n)$ denotes a dataset with n cases where each case d_i is described by a sequence of m words $d_i = (w_1^i, w_2^i, \cdots, w_m^i)$, and is associated with three labels l_i in R^p, c_i in R^q and a_i in R^r, where p and q denote the size of the one-hot vector of law and charge, while a_i is a multi-hot vector of articles with dimension r.

Each case d_i is also associated with a vector of p laws, $l_i = (l_1^i, l_2^i, \cdots, l_p^i)$, a vector of q articles, $a_i = (a_1^i, a_2^i, \cdots, a_q^i)$, and a vector of r charges, $c_i = (c_1^i, c_2^i, \cdots, c_r^i)$, where p, q, r represent the size of the three vectors and l_j^i, a_j^i, c_j^i in {0, 1}.

4. Proposed Models

Current models like Lawformer and TopJudge, as well as other state-of-the-art Legal Judgment Prediction (LJP) models, showcase the potential of neural network models in terms of accuracy and efficiency in predicting legal judgments. However, it is important to acknowledge that these models have certain limitations when applied to legal systems of different countries. For example, Lawformer is a pre-trained language model that utilizes legal documents from Mainland China as training data. It has shown impressive performance on the CAIL dataset. However, when applied to the TWLJP dataset, its performance is not as good as Chinese BERT. The reason could be attributed to variations in legal terminology, penalties, and writing styles of legal documents across different countries.

As mentioned before, Parameter Efficient Fine-Tuning (PEFT) is an alternative approach that allows a model to learn a new task with minimal updates. In PEFT, a pre-trained model is fine-tuned by selectively updating or adding a small number of parameters. Recent advancements in PEFT techniques have demonstrated the ability to achieve performance comparable to fine-tuning the entire model while only modifying a fraction (e.g., 0.01%) of its parameters [23].

In this paper, we adopt **LoRA**[10] to reduce the number of trainable parameters by learning pairs of rank-decomposition matrices while keeping the original weights frozen. The idea behind LoRA is that when adapting a pre-trained language model to a specific task or dataset, only a few features need to be emphasized or re-learnt. This means that the update matrix (ΔW) can be a low-rank matrix. As shown in Figure 2, The update of a pre-trained weight matrix $W \in R^{d \times k}$ is constrained by using a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in R^{d \times r}$, $A \in R^{r \times k}$, and the rank r is a hyper parameter less than or equal to the minimum of d and k. During training, W_0 remains unchanged and does not receive any gradient updates, while A and B contain trainable parameters.

Since our LJP tasks include the prediction of legal law, article, and charges, we add three fully connected layers to the CLS output as depicted in Figure 2. By adopting this approach with PLM frozen, we can significantly minimize the computational resources and time required for fine-tuning while ensuring the model's performance is preserved.

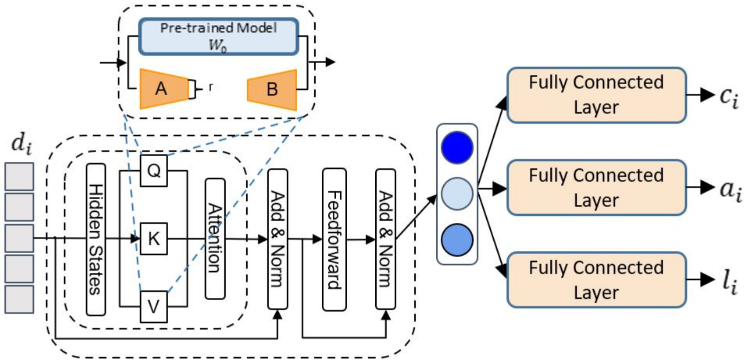

Figure 2: Model Architecture

The key advantage of **LoRA** lies in its remarkable ability to substantially reduce the computational resources and time necessary for fine-tuning, while maintaining the model's performance. This method proves particularly valuable when tackling extensive fine-tuning tasks, such as the refinement of highly capable large models that consist of billions of parameters.

5. Experiment

In order to evaluate the performance of the TWLJP dataset that we have collected across different pre-trained language models, we conducted training and evaluation using the following settings:

Multi-task BERT As dipicted in Figure2, we use multi-task learning to model the prediction of Law, Charge and Article by given criminal fact descriptions in the indictment as input. We utilize the Huggingface[24] Chinese pre-training language model bert-base-chinese. The optimizer we use for Multi-task BERT is BERT Adam with a learning rate of 1e-5, maximum length of 512 and hidden size of 768 for the parameters of pre-trained language model.

Multi-task Lawformer Lawformer[12] is a pre-trained language model based on the CAIL-long dataset and capable of processing articles up to 4096 characters in length. However, since Lawformer uses the CAIL-long dataset in simplified Chinese, and our data is in traditional Chinese, we first used the OpenCC package to convert the crime facts to simplified Chinese before training. The optimizer we use for Multi-task Lawformer is AdamW with a learning rate of 1e-5, maximum length of 512 and hidden size of 768 for the parameters of pre-trained language model.

LoRA We utilized the LoRA implementation from Hugging Face's PEFT package[25] and bert-based-chinese model to generate embeddings. In the LoRA setting, the value of r is set to 8. The optimizer we use for LoRA is AdamW with a learning rate of 3e-4, maximum length of 512 and hidden size of 768 for the parameters of pre-trained language model.

Evaluation Metric We adopt micro precision (MiP), recall (MiR), and F1 score (MiF), as well as macro precision (MaP), recall (MaR), and F1 score (MaF), as the evaluation metrics. Macro-precision/recall/F1 is computed by averaging each class, which is a commonly used metric in multi-label classification tasks.

Sub-task	Law					
Model/Metric	MiP	MiR	MiF	MaP	MaR	MaF
Multi-task BERT	**99.46±0.1**	99.04±0.1	99.24±0.1	96.2±1.0	93.28±2.0	94.52±0.7
Multi-task Lawformer	99.30±0.1	98.46±0.3	98.88±0.1	95.0±2.7	87.02±5.6	89.98±3.2
LoRA(r=8)	99.43±0.1	**99.10±0**	**99.27±0.1**	**96.50±0.5**	**93.87±1.6**	**95.03±1.0**

Table 3
The performance of Law prediction on TWLJP dataset. Mi means Micro, Ma means Macro.

Sub-task	Article					
Model/Metric	MiP	MiR	MiF	MaP	MaR	MaF
Multi-task BERT	**96.76±0.3**	94.60±0.7	95.68±0.3	80.60±5.1	72.20±2.9	74.6±3.7
Multi-task Lawformer	95.60±0.6	91.88±1.0	93.72±0.3	73.50±3.7	62.94±2.0	65.9±2.2
LoRA(r=8)	96.63±0.1	**95.10±0**	**95.87±0.1**	**84.90±4.7**	**78.07±2.3**	**80.23±3.4**

Table 4
The performance of Article prediction on TWLJP dataset. Mi means Micro, Ma means Macro.

Sub-task	Charge					
Model/Metric	MiP	MiR	MiF	MaP	MaR	MaF
Multi-task BERT	94.08±0.2	93.46±0.3	93.74±0.1	69.36±3.5	64.14±2.6	65.10±2.7
Multi-task Lawformer	93.00±1.0	92.46±0.2	92.76±0.5	64.44±3.2	59.14±2.5	59.94±1.6
LoRA(r=8)	**94.53±0.2**	**93.53±0.1**	**94.00±0**	**71.10±1.6**	**65.17±1.9**	**66.93±1.6**

Table 5
The performance of Charge prediction on TWLJP dataset. Mi means Micro, Ma means Macro.

5.1. Performance on TWLJP

To evaluate the performance of the TWLJP dataset across different models, we conducted experiments using the models introduced in the previous section. The performance of TWLJP on each model is shown in Tables 3, 4, and 5. In each experiment, we selected the epoch with the best performance on the validation dataset and tested on the testing dataset. The performance shown in the tables is the average performance of the model over five experiments, with a calculation of 2 times the standard deviation.

We conducted the experiments using the GeForce RTX 4070 Ti graphics card, and the training time for each model for one epoch, as well as the parameter information of the models, are presented in Table 6.

TWLJP	Multi-task BERT	Multi-task Lawformer	LoRA
Time	3hrs 49mins	3hrs 44mins	1hr 58mins
# Parameters	102,716,744	105,470,792	103,011,656
# Trainable Parameters	102,716,744	105,470,792	744,008

Table 6
The training time and parameter information for each model on the TWLJP dataset.

Based on the experimental results, it is evident that the performance of models implemented using the Lawformer pre-trained language model did not meet our expectations. Upon analysis, we determined that the reason behind this discrepancy lies in the fact that Lawformer was trained on legal documents from mainland China. Despite our efforts to convert the input criminal facts from Traditional Chinese to Simplified Chinese, there are significant differences between the legal systems and terminologies used in mainland China and Taiwan. This mismatch in legal terminology and usage negatively impacted the performance of Lawformer on the TWLJP dataset.

Under the training architecture of LoRA, comparable performance to Multi-task BERT is achieved in terms of case cause, legal provisions, and legal sources, and even superior performance compared to Multi-task BERT. The training time for one epoch is 1 hour and 58 minutes, which is approximately half the time required by Multi-task BERT, which is 3 hours and 49 minutes. Regarding the parameter count, Multi-task BERT has a total of 102,716,744 parameters, all of which need adjustment. In the LoRA architecture, the total number of parameters is 103,011,656, but only 744,008 parameters need to be trained, which is approximately 0.72% of the trainable parameters in Multi-task BERT.

Sub-task	Article					
Model/Metric	MiP	MiR	MiF	MaP	MaR	MaF
Multi-task BERT	84.1	85.7	84.9	79.0	**71.6**	73.4
Multi-task Lawformer	79.3	79.8	79.6	70.9	59.6	62.7
LoRA(r=8)	**84.6**	**86.9**	**85.8**	**79.9**	**71.6**	**73.9**

Table 7
The performance of Article prediction on CAIL dataset. Mi means Micro, Ma means Macro.

5.2. Performance on CAIL

To ensure fairness in our experiments, we also utilized the publicly available CAIL dataset. We conducted multi-task training on the dataset, focusing on the charges and articles. The perfor-

Sub-task	Charge					
Model/Metric	MiP	MiR	MiF	MaP	MaR	MaF
Multi-task BERT	**89.0**	**89.1**	**89.0**	84.4	**77.1**	**79.4**
Multi-task Lawformer	82.4	81.9	82.1	74.5	62.1	65.5
LoRA(r=8)	**89.0**	88.3	88.7	**84.8**	76.7	79.1

Table 8
The performance of Charge prediction on CAIL dataset. Mi means Micro, Ma means Macro.

mance of each model is shown in Tables 7 and 8. Since the main objective of our experiment was to compare the performance, time, and parameters of large language models, we did not compare them to other related models. For each experiment, we selected the epoch with the best performance on the validation dataset and tested it on the test dataset. We conducted the experiments using the GeForce RTX 4070 Ti graphics card, and the training time for each model per epoch and the parameter information are provided in Table 9.

TWLJP	Multi-task BERT	Multi-task Lawformer	LoRA
Time	2hrs 19mins	2hrs 11mins	1hr 14mins
# Parameters	102,859,778	105,613,826	103,154,690
# Trainable Parameters	102,859,778	105,613,826	887,042

Table 9
The training time and parameter information for each model on the CAIL dataset

From the experimental results, it can be observed that the performance of the Lawformer pretrained language model did not meet expectations. Upon analyzing the reasons for this, although Lawformer was trained on legal documents from mainland China, it is based on the Longformer architecture, which allows for input lengths of up to 4096 tokens. However, we used a maximum length of 512 tokens, and modifying this maximum length would lead to insufficient memory on the graphics card. As a result, the weights of some models were not updated, leading to poor training performance.

On the other hand, under the training framework of LoRA, comparable performance to Multi-task BERT was achieved for charges and articles. The training time for one epoch was 1 hour and 14 minutes, compared to 2 hours and 19 minutes for Multi-task BERT, requiring approximately half the time. In terms of parameter quantity, LoRA only required approximately 0.86% of the training parameters compared to Multi-task BERT.

6. Conclusion

Legal AI plays a crucial role in providing legal knowledge services to individuals with legal backgrounds, as well as assisting non-legal professionals. However, due to the diverse range of sub-tasks in Legal AI and the increasing size of pre-trained language models, training and storing a separate language model for each sub-task can be costly and resource-intensive. To address this challenge, we have embraced the concept of Parameter Efficient Fine-Tuning (PEFT) and applied it to the field of Legal AI.

By leveraging the PEFT approach, specifically through the implementation of the LoRA architecture, we have observed promising results in fine-tuning pre-trained language models. This approach allows us to achieve comparable, if not superior, performance while significantly reducing the time required for model adjustments. In our experiments, we found that using the LoRA framework required only about half the time compared to fine-tuning the entire model, without sacrificing performance. This innovative methodology opens up new possibilities for adapting language models to different legal contexts efficiently.

The success of our approach highlights the potential of PEFT techniques in the Legal AI domain. By efficiently adjusting and fine-tuning language models, we can tailor them to specific legal frameworks, taking into account the variations in legal definitions, documents, and terminologies across different countries. This advancement not only enhances the accuracy and relevance of legal knowledge services but also extends the accessibility of Legal AI to individuals without a legal background.

References

[1] H. Chen, L. Wu, J. Chen, W. Lu, J. Ding, A comparative study of automated legal text classification using random forests and deep learning, Information Processing & Management 59 (2022) 102798. URL: https://www.sciencedirect.com/science/article/pii/S0306457321002764. doi:10.1016/j.ipm.2021.102798.

[2] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, J. Xu, Cail2018: A large-scale legal dataset for judgment prediction, 2018. URL: https://arxiv.org/abs/1807.02478. doi:10.48550/ARXIV.1807.02478.

[3] Y. Lyu, Z. Wang, Z. Ren, P. Ren, Z. Chen, X. Liu, Y. Li, H. Li, H. Song, Improving legal judgment prediction through reinforced criminal element extraction, Information Processing & Management 59 (2022) 102780. URL: https://www.sciencedirect.com/science/article/pii/S0306457321002600. doi:10.1016/j.ipm.2021.102780.

[4] H. Ye, X. Jiang, Z. Luo, W. Chao, Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 1854–1864. URL: https://aclanthology.org/N18-1168. doi:10.18653/v1/N18-1168.

[5] L. Ma, Y. Zhang, T. Wang, X. Liu, W. Ye, C. Sun, S. Zhang, Legal judgment prediction with multi-stage case representation learning in the real court setting, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 993–1002. URL: https://doi.org/10.1145/3404835.3462945. doi:10.1145/3404835.3462945.

[6] Z. Hu, X. Li, C. Tu, Z. Liu, M. Sun, Few-shot charge prediction with discriminative legal attributes, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 487–498. URL: https://aclanthology.org/C18-1041.

[7] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, M. Sun, Legal judgment prediction via topological learning, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3540–3549. URL: https://aclanthology.org/D18-1390. doi:10.18653/v1/D18-1390.

[8] N. Xu, P. Wang, L. Chen, L. Pan, X. Wang, J. Zhao, Distinguish confusing law articles for legal judgment prediction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 3086–3095. URL: https://aclanthology.org/2020.acl-main.280. doi:10.18653/v1/2020.acl-main.280.

[9] Q. Bao, H. Zan, P. Gong, J. Chen, Y. Xiao, Charge prediction with legal attention, in: Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I, Springer-Verlag, Berlin, Heidelberg, 2019, p. 447–458. URL: https://doi.org/10.1007/978-3-030-32233-5_35. doi:10.1007/978-3-030-32233-5_35.

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[11] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, M. Sun, How does NLP benefit legal system: A summary of legal artificial intelligence, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5218–5230. URL: https://aclanthology.org/2020.acl-main.466. doi:10.18653/v1/2020.acl-main.466.

[12] C. Xiao, X. Hu, Z. Liu, C. Tu, M. Sun, Lawformer: A pre-trained language model for chinese legal long documents, AI Open 2 (2021) 79–84. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000176. doi:10.1016/j.aiopen.2021.06.003.

[13] N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, V. Lampos, Predicting judicial decisions of the european court of human rights: A natural language processing perspective, PeerJ Computer Science 2 (2016) e93.

[14] I. Chalkidis, I. Androutsopoulos, N. Aletras, Neural legal judgment prediction in English, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4317–4323. URL: https://aclanthology.org/P19-1424. doi:10.18653/v1/P19-1424.

[15] R. A. Shaikh, T. P. Sahu, V. Anand, Predicting outcomes of legal cases based on legal factors using classifiers, Procedia Computer Science 167 (2020) 2393–2402. URL: https://www.sciencedirect.com/science/article/pii/S1877050920307584. doi:10.1016/j.procs.2020.03.292, international Conference on Computational Intelligence and Data Science.

[16] L. Kang, J. Liu, L. Liu, D. Ye, Label definitions augmented interaction model for legal charge prediction, in: Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 –April 1, 2021, Proceedings, Part I, Springer-Verlag, Berlin, Heidelberg, 2021, p. 270–283. URL: https://doi.org/10.1007/978-3-030-72113-8_18. doi:10.1007/978-3-030-72113-8_18.

[17] B. Luo, Y. Feng, J. Xu, X. Zhang, D. Zhao, Learning to predict charges for criminal cases with legal basis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Den-

mark, 2017, pp. 2727–2736. URL: https://aclanthology.org/D17-1289. doi:10.18653/v1/D17-1289.

[18] L. Gan, K. Kuang, Y. Yang, F. Wu, Judgment prediction via injecting legal knowledge into neural networks, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2021) 12866–12874. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17522.

[19] W. Yang, W. Jia, X. Zhou, Y. Luo, Legal judgment prediction via multi-perspective bi-feedback network, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 4085–4091. URL: https://doi.org/10.24963/ijcai.2019/567. doi:10.24963/ijcai.2019/567.

[20] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: International Conference on Machine Learning, PMLR, 2019, pp. 2790–2799.

[21] X. L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 4582–4597. URL: https://aclanthology.org/2021.acl-long.353. doi:10.18653/v1/2021.acl-long.353.

[22] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, G. Neubig, Towards a unified view of parameter-efficient transfer learning, in: ICLR, 2022.

[23] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, C. A. Raffel, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, Advances in Neural Information Processing Systems 35 (2022) 1950–1965.

[24] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[25] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, Peft: State-of-the-art parameter-efficient fine-tuning methods, https://github.com/huggingface/peft, 2022.