Abstract

The generative and reasoning capabilities of Pre-trained Language Models (PLMs) have led to prominent advancements in the field of Natural Language Understanding (NLU). However, PLMs face two main challenges. Firstly, PLMs struggle to generalize on specialized domains such as Legal, Finance, and Medicine, due to the inherent characteristics of the domain-specific texts, including domain-specific vocabulary and complex sentence structure. Secondly, deploying PLMs in real-world applications pose significant challenges due to their high memory requirements and compute-intensive nature, rendering them unsuitable for many practical applications. In this work, we introduce a novel approach called Keyword Knowledge Distillation (KKD) for in-domain pre-training using Knowledge Distillation (KD). Instead of randomly masking tokens during KD training, KKD involves selectively masking in-domain keywords to preserve hints about domain-specific information and avoid unnecessary randomness. By employing KKD, we transfer the knowledge encoded in a large domain-specific Teacher BERT model to a smaller Student BERT model, which can then be fine-tuned to achieve strong performance across down streaming tasks.

We evaluate KKD on two specialized domains namely Legal and Finance. Using KKD, we have demonstrated that the capabilities of the larger Teacher models, Legal-BERT and Fin-BERT, can be retained in smaller Student models. We have preserved approximately 99.7% of Legal-BERT's capabilities and nearly 100% of Fin-BERT's capabilities in the Student models, which are 40% smaller and 60% faster. The Student model of Legal-BERT achieved an average micro-F1 score of 83.5% on the LexGLUE benchmark, while the Student model of Fin-BERT obtained 97% accuracy on the Financial PhraseBank dataset and a Mean Square Error (MSE) loss of 0.072 on FiQA sentiment scoring dataset.

1. Introduction

Over the past few years, Pre-trained Language Models (PLMs) have demonstrated exceptional performance across a range of NLP benchmarks. For example, GLUE [27], SQUAD [28], and RACE [29], have achieved near-human levels of proficiency. Despite their success in general NLP tasks, PLMs often underperform in specialized domains such as Legal, Finance, and Medicine [6,30,44,45,46]. This lower performance is mainly attributed to the unique characteristics of these domains, which differ significantly from general language. These differences include specialized vocabulary, intricate morphology, complex syntax, and domain-specific semantics [33,34,47,48,49]. These characteristics pose a significant challenge for PLMs, as they are trained on natural language corpora that may not adequately capture the intricacies of domain-specific text. Additionally, deploying PLMs in real-world applications presents significant challenges, primarily due to their size and resource demands. These models usually have a vast number of parameters, leading to high memory requirements and compute-intensive operations [37,38,39]. This poses a major obstacle for practical deployment. Furthermore, the large size of PLMs results in long inference times, further reducing their suitability for many real-world applications. Therefore, it is necessary to address the challenges of domain adaptation as well as the computational overhead of these PLMs. Several techniques, such as quantization [40], weights pruning [41], and Knowledge Distillation (KD) [42], have been developed for tackling these challenges. Prior work has shown that KD tends to achieve a better balance between compression and accuracy as compared to other model compression techniques [52]. Therefore, in this paper, we concentrate on KD. Specifically, we explore the application of KD in specialized domains, an area where this method has not been extensively studied. KD involves transferring knowledge from a large, pre-trained Teacher model to a smaller Student model, effectively condensing the knowledge of the former into the latter. By leveraging KD in these specialized domains, we aim to bridge the gap between the computational demands of LLMs and the practical constraints of real-world applications in domain-specific text processing.

We propose Keyword Knowledge Distillation (KKD), a novel approach for domain-specific pre-training of PLMs. KKD facilitates the transfer of domain-specific knowledge from a large domain-specific pre-trained BERT Teacher model (110M parameters) to a smaller Student BERT model (66M parameters). We use domain-specific BERT models because they are pre-trained on a large corpus of text relevant to their respective fields, allowing them to accurately capture domain-specific vocabulary and nuances. KKD proposes a selective masking strategy that focuses on masking in-domain keywords during model training. This strategy utilises KEYBERT [14] to identify contextually relevant keywords. It identifies the top 15% of the keywords in the training corpus based on their relevance score and masks them during the Student model's KD training for the masked language modelling objective. This targeted approach preserves hints about domain-specific information and helps in mitigating randomness. We utilise KKD on two specialized domains namely Legal and Finance. Our findings suggest that KKD offers a promising avenue for effectively distilling domain-specific knowledge into smaller models, thereby enhancing their utility in real-world domain-specific text processing applications.

To summarize, we make the following contributions in this work:

1. We propose Keyword Knowledge Distillation (KKD), a novel KD-based pre-training approach specifically designed for domain-specific tasks. KKD involves in domain knowledge transfer from a large Teacher model to a smaller Student model. The compressed Student model can then be fine-tuned for various domain-specific tasks.
2. . Through an extensive set of experiments, we show that KKD significantly improves the computational efficiency of domain-specific PLMs. By employing KKD, we have shown that the capabilities of the larger Teacher models, Legal-BERT and Fin-BERT, can be effectively transferred to smaller Student models. These Student models retain approximately 99.7% of Legal-BERT's capabilities and nearly 100% of Fin-BERT's capabilities, while being 40% smaller and 60% faster.

3. We conduct a detailed ablation study to show that KKD preserves hints about domain-specific information and surpasses the standard KD approach on specialized tasks.

The rest of the paper is organised as follows: Section 2 elaborates on the necessary background and related work. Section 3 presents our proposed method. Section 4 details out the dataset and experimental setup used in this work. Section 5 discusses the results obtained and presents the ablation study. Section 6 concludes the paper and provides directions for future work.

2. Background and Related Work

2.1. Knowledge Distillation

Knowledge Distillation (KD) is a compression technique where a compact model, referred to as the Student model, is trained to replicate the behaviour of a larger model [1], which can be either a single Teacher model or an ensemble of models [2]. In case of PLMs, KD involves transferring the knowledge from a complex, pre-trained Teacher model to a smaller, more lightweight Student model [3,4,5]. This process allows the Student model to achieve a performance at par with the Teacher model while being more computationally efficient.

Let f_t and f_s represent the Teacher and Student models, respectively. These models transform inputs into informative representations called logits. The training objective is formulated such that the Student model learns to mimic the Teacher model by producing similar output logits. If the Student model can replicate the Teacher model's logits, it should ideally perform comparably to the Teacher model on downstream tasks. Mathematically, KD can be represented by the minimization of the objective function as shown in Equation-1:

$$L_{kd} = L\ (\ f_t(x),f_s(x)) \qquad (1)$$

where L(·) is a loss function that measures the disparity between the logits of the Teacher and Student models during the training process, x represents the text input, and refers to the training dataset. During training, the Teacher model is kept frozen, and the Student model attempts to minimise the objective loss. This process allows the model to distil the knowledge of the Teacher model.

2.2. Related Work

KD stands out as a widely employed method for compressing models. We explored some of the recent advancements in KD techniques, specifically in the realm of Language Models (LMs). Tang et al. [16] proposed a method to distil knowledge from BERT into a single-layer Bi-LSTM. DistilBERT proposed by Sanh et al. [17] is a smaller, faster version of the BERT model. In the distillation process, a large BERT model is used as a Teacher model to train the smaller DistilBERT model which demonstrates competitive performance across various NLP tasks. Sun et al. [18] used Patient Knowledge Distillation (PKD), which utilises knowledge from multiple intermediate layers of the large model during training. Jiao et al. [19] proposed a two-stage learning framework that ensures that TinyBERT captures both general-domain and task-specific knowledge from BERT. FastBERT introduced by Liu et al. [20] addresses the computational inefficiency of large language models like BERT. Liang et al [21] introduced MixKD, a data-agnostic distillation framework that utilises mixup, a simple data augmentation approach, to improve the generalisation ability of Student models. Wang et al. [22] proposed MINILM, a method for mimicking the Teacher model's self-attention modules, particularly the scaled dot-product between values. They further extend their work in MINILMv2 [23] by defining multi-head self-attention relations and using them to train a Student model, allowing for flexibility in the number of attention heads compared to the previous method. The work by Zhang et al. [24] proposes PTLoss, a novel distillation objective, to improve knowledge distillation by addressing discrepancies between the Teacher model's output distribution and the ground truth. Wu et al. [25] introduced an attribution-driven KD by exploring token-level rationale using Integrated Gradients (IG) and employing multi-view attribution distillation for enhancing the knowledge transfer. MINILLM introduced by Gu et al. [26] reverse Kullback-Leibler divergence (KLD) in place of forward KLD to prevent overestimation of low-probability regions by the Student model and demonstrate superior performance of their approach.

All the existing KD methods discussed above have been extensively evaluated on general-purpose English datasets, demonstrating their effectiveness in transferring knowledge from large Teacher models to smaller Student models. However, to the best of our knowledge, KD in the context of domain-specific datasets such as Legal and Finance is an unexplored area. To address this limitation, we conducted an evaluation of standard KD, using the output distributions of the Teacher model and the Student model as the objective function. Our findings reveal that while the standard KD method provides valuable insights, it is limited in its ability to effectively transfer knowledge in specialized domains. In contrast, our proposed KD method is tailored for specialized domains. This approach not only fills a crucial gap in the literature but also underscores the importance of domain-specific considerations in knowledge distillation techniques.

3. Method

In this section, we detail the proposed method. We first delve into Distilled-BERT, which lays the groundwork for understanding our approach. Building upon this foundation, we then introduce our novel method, KKD, which addresses the limitations of traditional Masked Language Modeling (MLM) in domain-specific contexts such as legal and financial text.

3.1 Distilled Bert

In this work, we build upon the underlying KD objective function used for training the Distilled-BERT model [3], because it provides a comprehensive framework for training the Student model by incorporating various loss components, including distillation loss L_{kd}, MLM loss L_{mlm} [7], and cosine embedding loss L_{cos}. This holistic approach ensures effective knowledge transfer from the Teacher model to the Student model. According to [3], the final objective function is a linear combination L_{kd}, L_{mlm}, and L_{cos}. The Student model is trained to minimise the objective function. 3.2 Keyword Knowledge Distillation (KKD)

As mentioned in Section 3.1, the KD objective utilized in the training of Distilled-BERT incorporates an MLM loss term. MLM involves the random masking of tokens within the input sequence and training the model to predict the masked tokens [7]. However, the application of MLM in the pretraining of LMs on domain-specific data presents notable challenges. In case of legal or financial text, certain tokens are more critical than others for a comprehensive understanding. Randomly masking such tokens can lead to the model focusing on less important tokens, resulting in suboptimal results [43]. Furthermore, these texts are characterized by specialized vocabulary and complex syntactic structures that may not be sufficiently captured by traditional MLMs. This inherent limitation may result in the model learning irrelevant patterns, thus hindering its ability to effectively learn domain-specific characteristics. For instance, legal documents frequently contain terms such as “plaintiff,” “defendant,” “judgment,” and “appeal,” which are pivotal for understanding the overall context. If the model randomly masks tokens during MLM training, it risks overemphasizing less critical tokens and stop words, which, despite their higher frequency, are less informative in the legal domain. It can ultimately hamper the learning of crucial domain-specific patterns.

3.2.1 Keyword Masking

To address the above mentioned challenges, we employed a domain adaptation-based approach for pre-training LMs that relies exclusively on masking in-domain keywords. Initially, we extracted contextually relevant keywords from each document and selected the most relevant keywords to be masked during the training phase. To identify in-domain keywords, we incorporated KeyBERT [14], a transformer-based model designed for keyword extraction. It utilizes BERT embeddings to score words or phrases based on their relevance to a given text. In this work, we used KeyBERT to identify and extract the top 15% most relevant keywords that are crucial for understanding domain-specific data. We chose the 15% threshold because it is a standard percentage for mask tokens in BERT, ensuring a balance between comprehensiveness and focus. After identifying the most relevant keywords, we masked them and calculated the Keyword MLM loss L_{kmlm}. Figure 1 shows how keyword masking enables the model to focus on contextually relevant words in the context of legal domain.

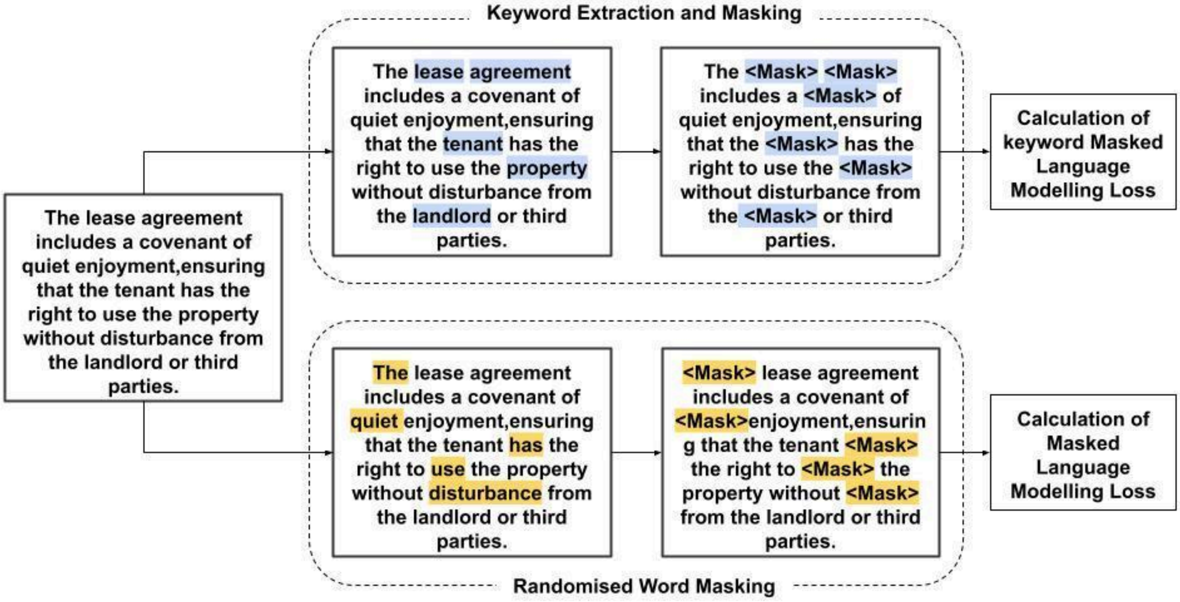
Keyword Extraction and Masking
The lease agreement includes a covenant of quiet enjoyment,ensuring that the tenant has the right to use the property without disturbance from the landlord or third parties.
The lease agreement includes a covenant of quiet enjoyment,ensuring that the tenant has the right to use the property without disturbance from the landlord or third parties.
The <Mask> <Mask> includes a <Mask> of quiet enjoyment,ensuring that the <Mask> has the right to use the <Mask> without disturbance from the <Mask> or third parties.
Calculation of keyword Masked Language Modelling Loss
The lease agreement includes a covenant of quiet enjoyment,ensuring that the tenant has the right to use the property without disturbance from the landlord or third parties.
<Mask> lease agreement includes a covenant of <Mask>enjoyment,ensurin g that the tenant <Mask> the right to <Mask> the property without <Mask> from the landlord or third parties.
Calculation of Masked Language Modelling Loss
Randomised Word Masking

Fig-1: Domain-specific Keyword Masking vs Randomized Word Masking

3.2.2 Workflow

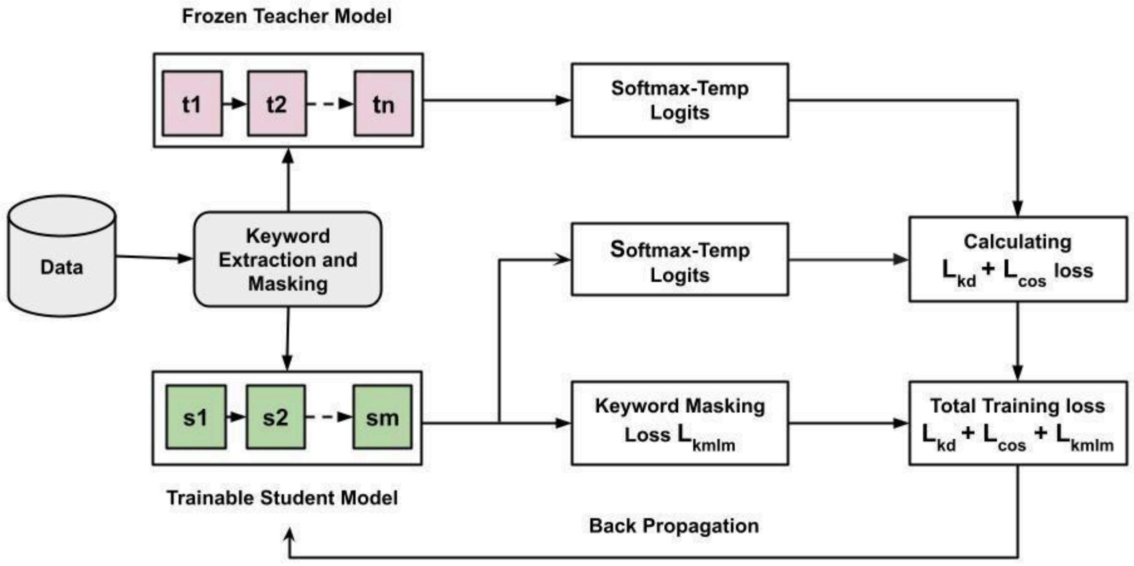
Frozen Teacher Model
t1
t2
tn
Softmax-Temp Logits
Data
Keyword Extraction and Masking
Softmax-Temp Logits
Calculating $L_{kd} + L_{cos}$ loss
s1
s2
sm
Keyword Masking Loss L_{kmlm}
Total Training loss $L_{kd} + L_{cos} + L_{kmlm}$
Trainable Student Model
Back Propagation

Fig-2: Overall Workflow of KKD

The overall workflow of KKD is depicted in Figure 2. The input text is processed by converting it into tokens before being passed to both the Teacher and Student models. During this step, the Teacher model is frozen, which means that its parameters are not updated during training. On the other hand, the Student model is trainable, and its parameters are updated to minimise the loss. Before feeding the tokens into the Student model, the top 15% relevant tokens are masked out using the keyword masking strategy elaborated in Section 3.2.1. The Student model then predicts the masked tokens based on the context provided by the surrounding tokens. The L_{kmlm} loss is calculated based on the predictions of the Student model with respect to the masked tokens. In addition, both the Teacher and Student models generate output logits. The final training loss L_f is a linear combination of the distillation loss L_{kd} (in our case we have used KL-Divergence loss as L_{kd}), the keyword MLM loss $L_{kmlm,}$ and the cosine embedding loss L_{cos} . The final training loss is defined in equation 2.

$$L_f = L_{kd} + L_{cos} + L_{kmlm} \qquad (2)$$

The L_{kd} loss and the L_{cos} loss is calculated using the softmax-temperature: pi = $\exp(z_{i/}T)$, where z_i is the ith logits and T is the smoothness factor over the raw logits from both the Teacher and Student models. Algorithm 1 provides a pseudo-code illustrating the workflow of KKD.

Algorithm 1: Pseudo code delineating the implementation of KKD

```
def keyword_knowledge_distilation(input_text):

    # Tokenizing the input sentence
    tokens = tokenize(input_text)

    # mask_top_tokens function indentifies the top 15% import tokens in the input sentence
    masked_tokens = mask_top_tokens(tokens, 15)

    # The output logits are generated from the Teacher and the Student Model
    Teacher_logits = Teacher_model(masked_tokens)
    Student_logits = Student_model(masked_tokens)

     #Applying softmax temperature to the logits from the Teacher and the Student model
    Teacher_softmax = softmax(Teacher_logits / temperature)
    Student_softmax = softmax(Student_logits / temperature)

    #Calculating the KL-Divergence loss
    kd_loss = kl_divergence(Student_softmax, Teacher_softmax)

    #Calculating the Keyword Masking loss
    kmlm_loss = keyword_masked_loss(Student_model, masked_tokens)

    #Calculating the Cosine Embedding loss
    cosine_loss = cosine_embedding_loss(Student_logits, Teacher_logits)

    #Calculating the Final loss
    total_loss = kd_loss + kmlm_loss + cosine_loss

    #The Student model tries to minimize the loss by updating its weights
    Student_model.update_parameters(total_loss)
```

4. Experimentation and Dataset Details

4.1 Legal domain

4.1.1 Teacher Model Architecture for Legal Domain

We used the Legal-BERT model introduced by Chalkidis et al. [6]. (2020) as the Teacher modelfor KKD. Legal-BERT was trained from scratch on a diverse English legal corpus totalling 12 GB of data. This corpus consists of various types of legal documents, including legislation, court cases, and contracts, sourced from publicly available resources. Legal-BERT shares the same architecture as BERT-BASE [7], featuring 12 layers, 768 hidden units, and 12 attention heads, amounting to 110M parameters.

4.1.2 Dataset Description for KKD training for Legal Domain

For training the Student model using KKD, we utilised the Pile of Law dataset [8], a curated collection of legal and administrative text that is currently around 256GB in size and expanding. The dataset comprises data from 35 different sources, including legal analyses, court opinions and filings, government agency publications, contracts, statutes, regulations, and casebooks. We selected the initial 1.2 million samples from this dataset, which provides a substantial number of samples to ensure robust training of the Student model while also requiring fewer computing resources and less experimentation time compared to using the entire dataset. Finally, we applied an 80-20 split for training and validation.

4.1.3 Downstream Tasks and Experimental Setup

We evaluated the results of KKD on the LexGLUE benchmark [15], a collection of seven datasets for evaluating model performance on a diverse set of legal Language Understanding (LLU) tasks. The description of each dataset is as follows:

1. ECtHR (Task A): It is a multi-label classification dataset in the legal domain. The task involves analysing factual descriptions of legal cases and identifying all applicable articles from the European Convention on Human Rights (ECHR) which have been violated.

2. ECtHR (Task B): Similar to ECtHR-A, it is also a multi-label classification dataset. Given factual summaries of legal cases, the task involves predicting potentially violated articles by the presented situation.

3. SCOTUS: It is based on legal topic classification in the US court system. This task falls under the category of single-label multi-class classification. Given a Supreme Court opinion as input, the model must predict the most relevant legal issue area the case pertains to.

4. EUR-LEX: It is designed for legal document annotation. This task centres on multi-label classification, where the goal is to predict the relevant EuroVoc concepts (legal categories) associated with a given piece of European Union (EU) legislation.

5. LEDGER: It is a dataset for contract provision classification. The contract provisions are collected from documents obtained from the US Securities and Exchange Commission (SEC). The dataset falls under the category of multi-label classification.

6. UNFAIR-ToS: This dataset is designed for the classification of unfair terms present in online Terms of Service (ToS) agreements. This task falls under the category of multi-class classification. The goal is to analyse individual sentences from ToS agreements and predict whether they can be classified into one of the eight predefined categories of unfair contractual terms.

7. Case-Hold: It is designed to evaluate a fundamental legal task of identifying the holding (legal conclusion) of a cited case. This task, presented as a multiple-choice question with five answer options, focuses on the citing context provided within a judicial decision.

We report the scores achieved on various tasks by fine-tuning the Student model (Legal-KKD-Student) on the corresponding datasets, using the same configuration as Chadwick et al. [15] to ensure a fair comparison. Similar to them, we run five repetitions with different random seeds and report the test scores based on the seed with the best scores on development data. We used the Adam optimizer with an initial learning rate of 3e-5 and the maximum input token length of 128, trained up to 20 epochs with early stopping based on development data. Following Chadwick et al.'s [15] approach, we employ a hierarchical variant of our model for tasks such as ECtHR (A and B) and SCOTUS, which is not specifically designed for longer texts during fine-tuning. This hierarchical model utilises a pre-trained Transformer-based model to encode each paragraph of the input text independently, obtaining top-level representations h[cls] for each paragraph. A second-level shallow (2-layered) Transformer encoder, consistent across all model specifications, is then used to make the paragraph representations context-aware. Finally, we max-pool over the context-aware paragraph representations to obtain a document representation, which is passed to a classification layer.

4.2 Financial domain

4.2.1 Teacher Model Architecture for Financial domain

We utilised Fin-BERT model proposed by Araci [44] as the Teacher model. Fin-BERT is essentially a BERT-BASE model which was pretrained on a financial corpus called TRC2-financial.

4.2.2 Dataset Description for KKD training for Finance Domain

For KKD training in the financial domain, we utilized Reuters corpus[1]. Similar to Legal domain we sampled 1.2 million instances and employed 80-20 split for training and evaluation.

4.2.3 Downstream Tasks and Experimental Setup

We assess the performance of KKD on two downstream tasks in the financial domain: classification and regression. For evaluation, we utilised the following datasets:

1. Financial PhraseBank [50]: It contains 4845 English sentences and designed to aid sentiment classification task in the financial domain. It consists of three sentiment classes: positive, negative, and neutral. The dataset also presents the agreement levels on the sentences by the annotators. The distribution of the agreement levels and the number of samples are mentioned in Table-1.

Agreement level	Number of Samples

[1] https://trec.nist.gov/data/reuters/reuters.html

100%	2262
75% - 99%	1191
66% - 74%	765
50% - 65%	627
ALL DATA	4845

Table 1: Distribution of the Dataset on Different Agreement Level

2. FiQA Sentiment [51]: It contains 1,174 financial texts like news headlines or social media posts related to finance. The task is to assign a sentiment score between [-1,1] to financial texts, with 1 being the most positive sentence. It uses continuous scores for finer-grained sentiment analysis, thus making it a regression task.

We report the scores obtained by fine-tuning the Student model (Fin-KKD-Student). Similar to Arac [44], we have used 10-fold cross validation with the learning rate of 2e – 5, batch size of 64, and a maximum input sequence length of 64 tokens for evaluating KKD. In our study, we used 10 epochs for the classification task, and 15 epochs for the regression task. We used cross-entropy loss function for the classification task and the mean squared loss function for the regression task.

4.3 Student Model Architecture

In this work, we kept the underlying architecture of the Student model same as that of the Teacher model, apart from the number of layers in the encoder. Decreasing the number of layers enabled us to reduce the number of parameters from 110M in the original Teacher model to 66M million in the Student model. Therefore, our Student model is a 6-layer model, having 768 hidden units, and 12 attention heads.

4.4 Hyperparameters used for KKD Training

We conducted our experiments using a batch size of 32, a learning rate of 0.00001, and a maximum sequence length of 512. The training was carried out over 10 epochs. All experiments were performed on an NVIDIA TESLA V100 GPU with 32 GiB of memory. The loss curves for the training process are depicted in Figure 3.

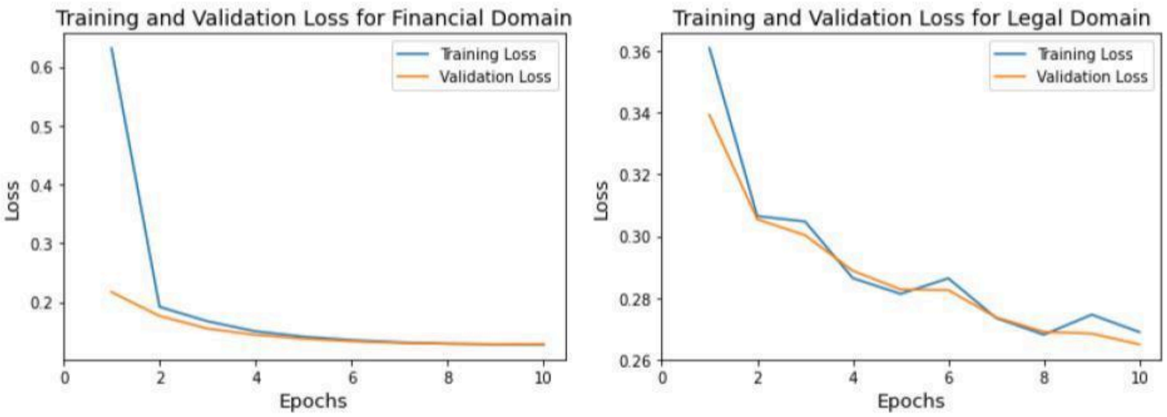

Fig-3: Loss Plot of KKD Training on Financial and Legal Domain

5. Experimental Results

5.1 Evaluation on Downstream Tasks

5.1.1 Legal Domain

We report the scores achieved by KKD on the LexGLUE benchmark datasets in Table 2. The table also compares the results of Legal-KKD-Student with other LM baselines reported in Chadwick et al. [15]. The performance based on micro-F1 (μ-F1) and macro-F1(m-F1) scores across all datasets is reported. Note that the Legal-BERT model reported in the table is the Teacher model used in KKD training, as explained in section 4.1.1. It is evident from Table-2 that the Legal-KKD-Student achieves results that are at par with the baseline models, even though there is a significant difference in model size. We observed the Legal-KKD-Student having 66M parameters performed equally well as compared to two legal domain-specific pre-trained models Legal-BERT and CaseLaw-BERT, which have 110M parameters. While considering the u-F1 score that provides a single, balanced measure of a model's overall performance across all classes. Table-3 reported the Aggregated Mean u-F1 and Aggregated Mean m-F1 score. Legal-KKD-Student obtained an Aggregated Mean u-F1 score of 80.0% as compared to the Teacher model's 79.8%. While in the case of m-f1, which considers performance disparities among individual classes, the Aggregated Mean m-F1 score for Legal-KKD-Student was 71.4%., which marginally lagged behind Legal-BERT's 72.0% score.

Method	Params	ECtHR (A)	ECtHR (B)	SCOTUS	EUR-LEX	LEDGER	UNFAIR-ToS	CaseHOLD	Aggregated Mean

		μ-F1	m-F1	μ-F1	m-F1	μ-F1	m-F1	μ-F1	m-F1	μ-F1	m-F1	μ-F1	m-F1	μ-F1/m-F1	μ-F1	m-F1
BERT	110 M	**71.2**	63.6	**79.7**	73.4	68.3	58.3	71.4	57.2	87.6	81.8	95.6	81.3	70.8	77.8	69.5
RoBERTa	125 M	69.2	59	77.3	68.9	71.6	62	71.9	57.9	87.9	82.3	95.2	79.2	71.4	77.8	68.7
DeBERTa	139 M	70	60.8	78.8	71	71.1	62.7	72.1	57.4	88.2	**83.1**	95.5	80.3	72.6	78.3	69.7
Longformer	149 M	69.9	64.7	79.4	71.7	72.9	64	71.6	57.7	88.2	83	95.5	80.9	71.9	78.5	70.5
BigBird	127 M	70	62.9	78.8	70.9	72.8	62	71.5	56.8	87.8	82.6	95.7	81.3	70.8	78.2	69.6
CaseLaw-BERT	110 M	69.8	62.9	78.8	70.3	76.6	65.9	70.7	56.6	**88.3**	83	96	82.3	**75.4**	79.4	70.9
Legal-BERT (Teacher Model)	110 M	70	64	80.4	**74.7**	76.4	66.5	72.1	57.4	88.2	83	**96**	**83**	75.3	79.8	72
Legal-KKD-Student	66M	70.8	**64.9**	78.6	73.3	**78.7**	**70.8**	**76.9**	**60.8**	88.2	82.3	95.9	76.2	70.6	**80**	**71.4**
Legal-KKD-Student w/o L_{kmlm}	66M	70.2	64	78.4	73	76.6	68.5	75.3	58.6	86.4	80.4	93.5	73.9	70.2	78.7	69.7
Legal-KKD-Student w/o L_{kmlm} w/o L_{cos}	66M	70.4	64.3	78.2	72.1	76.4	67.1	74.7	57.5	86.3	80.2	93.3	73.5	69.6	78.4	69.1

Table 2: Performance comparison of the proposed method with state-of-the-art model on the Lex-Glue benchmark tasks. All other results are reported in the work of Chadwick et al. [15] For each task, the best model is marked in **bold**; second best is underlined. All reported results are in percentage (%).

In Table-2 we also present an ablation experiment to determine the effectiveness of keyword masking technique utilised in our work. To perform this experiment, we conducted KD training without keyword masking. Therefore, we replaced the L_{kmlm} loss used in Legal -KKD with the standard L_{mlm} loss that involves randomly masking tokens in the input sequence. Subsequently, we also removed the L_{cos} loss from the KD process and evaluated the results. We observed that there is a decrease in performance without using keyword masking during KD training and results further decreased when we removed the L_{cos} loss term from KD training. The Aggregated μ-F1 score decreased from 80.0% to 78.7% and the Aggregated m-F1 score decreased from 71.4% to 69.7% when we removed the L_{kmlm} loss; it dipped a bit more from 78.7% to 78.4% in the case of Aggregated μ-F1 score and 69.7% to 69.1% for the case of Aggregated m-F1 score, when the L_{cos} loss factor was removed. Thus, the proposed method enhances the ability of the Student model to learn domain-specific patterns during KD training and improves results on LLU tasks.

5.1.2 Financial Domain

				ALL DATA			100% Agreement
Model	Params	m-F1	μ-F1	Accuracy	m-F1	μ-F1	Accuracy
Fin-BERT	110M	**84.0**	-	**86.0**	**97.0**	-	95.0

Fin-KKD-Student	66M	**84.0**	**85.0**	85.0	96.0	**97.0**	**97.0**
Fin-KKD-Student w/o L_{kmlm}	66M	81.0	83.0	83.0	93.0	94.0	95.0
KD-Student w/o L_{kmlm} w/o L_{cos}	66M	80.0	82.0	82.0	92.0	94.0	94.0

Table:3 Experimental Results on the Financial PhraseBank dataset. The Fin-BERT result is taken from the work of Araci [44], the best model is marked in **bold**; second best is underlined. All reported results are in percentage (%).

Table 3 presents the results for the classification task in the Financial Phrasebank dataset, comparing the performance of the Fin-KKD-Student with the baseline Teacher model, Fin-BERT, on both the entire dataset (ALL DATA) and a subset with 100% annotator agreement. In the case of the whole dataset, an m-F1 score of 84.0% is observed for Fin-KKD-Student similar to the m-f1 score of 84.0 % obtained by the Teacher model. The Student model achieved an Accuracy of 85.0% as compared to the Teacher model's 86.0%, showing that the Student model performs at par with the Teacher model, in spite of using 60% less number of parameters. For the subset of Financial Phrasebank dataset with 100% annotator agreement, the Student model obtained a m-f1 score of 96.0% and an Accuracy of 97.0% as compared to Teacher model's m-f1 score of 97.0% and the Accuracy score of 95.0%, demonstrating that the Student model obtained similar results as the Teacher model.

The ablation experiments for KKD are also mentioned in Table-3. Initially, we replaced L_{kmlm} in KKD with L_{mlm} and subsequently removed L_{cos} for further analysis. It is evident that when using both the entire dataset and a subset with 100% annotator agreement, there is a noticeable decrease in performance, which is further exacerbated when L_{cos} is removed. Specifically, the m-F1 score for the entire dataset decreases from 84.0% to 81.0% when using L_{mlm} . Similarly, for the subset with 100% annotator agreement, the m-F1 score decreases from 97.0% to 95.0% when L_{mlm} is used instead of $L_{kmlm.}$ Again a slight dip in results is observed when we omit the L_{cos} loss factor on both the entire dataset and a subset with 100% annotator agreement.

Model	Params	Mse	R^2
Fin-BERT	110M	0.07	0.55
Fin-KKD-Student	66M	0.07	0.54
Fin-KKD-Student w/o L_{kmlm}	66M	0.10	0.39
n-KKD-Student w/o L_{kmlm} w/o L_{cos}	66M	0.11	0.33

Table:4 Experimental Results on FiQA Sentiment Dataset. The Fin-BERT result is taken from the work of Araci [44], the best model is marked in **bold**; second best is underlined.

Table-4 presents the results of the regression task on the FiQA sentiment analysis dataset. The Student model achieved performance similar to that of the Teacher model, with an MSE of 0.07 and an R2 score of 0.53, compared to the Teacher model's MSE of 0.07 and R2 score of 0.55. This indicates that the Student model's performance is comparable to that of the Teacher model. Table-4 also reports the results of the ablation study for the regression task. The MSE score increases from 0.07 to 0.10 when L_{kmlm} is replaced with L_{mlm} and further rises to 0.11 when L_{cos} is removed. Similarly, the R^2 score declines from 0.54 to 0.39 with the replacement of L_{kmlm} with L_{mlm} and further decreases to 0.33 upon removing L_{cos}. These results highlight the importance of L_{kmlm} in enhancing performance. Hence our proposed method enhances the Student model's ability to learn domain-specific patterns during the KKD training for the financial domain.

5.2 Inference time

We have reported the inference time of both the Teacher and the Student model. The inference time for the legal domain models was measured by using EUR-LEX's validation set, the results for the Teacher model (Legal-BERT) and the Legal-KKD-Student model are reported in Table-5. The inference time for the financial models was measured by using Financial PhraseBank's validation set and the results for the Teacher model (Fin-BERT) and the Fin-KKD-Student model are mentioned in Table-6. All of these experiments were done using an Intel Xeon CPU with 2 vCPUs and 13GB of RAM, with batch size set to 1. From both of the tables, the Student model was found to be approximately 60% faster than the Teacher model. The significant improvement in inference time can be attributed to the difference in model size between the Teacher and the Student model.

Model	Inference Time (Milli Seconds)
Legal-BERT	2303.8
Legal-KKD-Student	1364.7

Table 5: Comparison of the Inference time on EUR-LEX validation dataset

Model	Inference Time (Milli Seconds)
Fin-BERT	2297.1
Legal-KKD-Student	1361.4

Table 6: Comparison of the Inference time on Financial PhraseBank validation dataset

6.Conclusion

In this work, we present a novel approach named Keyword Knowledge Distillation (KKD), for in-domain pre-training for specialized domains using KD. Our method addresses the challenges of deploying large PLMs in specialized domains by selectively masking for in-domain keywords, preserving domain-specific information, and reducing randomness. We distil knowledge from the large domain-specific Teacher BERT models to the compact Student BERT models. Importantly, it performs on par with the state of art domain-specific BERT models, while achieving significant parameter reduction (40%) and computational efficiency (60% faster)

In future work, we plan to explore the extraction of knowledge from intermediate layers of the Teacher model, in addition to the output distribution, as these layers are known to learn discriminative features that can benefit the training of the Student model.

6.References

1. Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network." *arXiv preprint arXiv:1503.02531* (2015).
2. Fukuda, Takashi, et al. "Efficient Knowledge Distillation from an Ensemble of Teachers." *Interspeech*. 2017.
3.Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).
4. Jiao, Xiaoqi, et al. "Tinybert: Distilling bert for natural language understanding." *arXiv preprint arXiv:1909.10351* (2019).
5. Gu, Yuxian, et al. "MiniLLM: Knowledge distillation of large language models." *The Twelfth International Conference on Learning Representations*. 2023.
6. Chalkidis, Ilias, et al. " LEGAL-BERT: The muppets straight out of law school." *arXiv preprint arXiv:2010.02559* (2020).
7.Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).
8. Henderson, Peter, et al. "Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset." *Advances in Neural Information Processing Systems* 35 (2022): 29217-29234.
9. Haigh, Rupert. *International legal English: A Practical Introduction for Students and Professionals*. Routledge, 2020.
10. Williams, Christopher. *Tradition and change in legal English: Verbal constructions in prescriptive texts*. Vol. 20. Peter Lang, 2007.
11. Sinsheimer, Ann. "Christopher Williams, Tradition and change in legal English: Verbal constructions in prescriptive texts." *Language in Society* 36.3 (2007): 473-474.
12. Ghosh, Sreyan, et al. "Dale: Generative data augmentation for low-resource legal nlp." *arXiv preprint arXiv:2310.15799* (2023).
13. Golchin, Shahriar, et al. "Do not Mask Randomly: Effective Domain-adaptive Pre-training by Masking In-domain Keywords." *arXiv preprint arXiv:2307.07160* (2023).
14. Maarten Grootendorst. 2020. Keybert: Minimal keyword extraction with bert
15. Chalkidis, Ilias, et al. "LexGLUE: A benchmark dataset for legal language understanding in English." *arXiv preprint arXiv:2110.00976* (2021).
16. Tang, Raphael, et al. "Distilling task-specific knowledge from bert into simple neural networks." arXiv preprint arXiv:1903.12136 (2019).
17. Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108 (2019).
18. Sun, Siqi, et al. "Patient knowledge distillation for bert model compression." arXiv preprint arXiv:1908.09355 (2019).
19. Jiao, Xiaoqi, et al. "Tinybert: Distilling bert for natural language understanding." arXiv preprint arXiv:1909.10351 (2019).

20. Liu, Weijie, et al. "Fastbert: a self-distilling bert with adaptive inference time." arXiv preprint arXiv:2004.02178 (2020).
21. Liang, Kevin J., et al. "Mixkd: Towards efficient distillation of large-scale language models." arXiv preprint arXiv:2011.00593 (2020).
22. Wang, Wenhui, et al. "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers." Advances in Neural Information Processing Systems 33 (2020): 5776-5788.
23. Wang, Wenhui, et al. "Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers." arXiv preprint arXiv:2012.15828 (2020).
24. Zhang, Rongzhi, et al. "Do not blindly imitate the Teacher: Using perturbed loss for knowledge distillation." arXiv preprint arXiv:2305.05010 (2023).
generation with pseudo-target training." arXiv preprint arXiv:2305.02031 (2023).
25. Wu, Siyue, et al. "Ad-kd: Attribution-driven knowledge distillation for language model compression." arXiv preprint arXiv:2305.10010 (2023).
26. Gu, Yuxian, et al. "MiniLLM: Knowledge distillation of large language models." The Twelfth International Conference on Learning Representations. 2023.
27. Wang, Alex, et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding." *arXiv preprint arXiv:1804.07461* (2018).
28. Rajpurkar, Pranav, et al. "Squad: 100,000+ questions for machine comprehension of text." *arXiv preprint arXiv:1606.05250* (2016).
29. Lai, Guokun, et al. "Race: Large-scale reading comprehension dataset from examinations." *arXiv preprint arXiv:1704.04683* (2017).
30. Zheng, Lucia, et al. "When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings." *Proceedings of the eighteenth international conference on artificial intelligence and law*. 2021.
31. Elwany, Emad, Dave Moore, and Gaurav Oberoi. "Bert goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding." *arXiv preprint arXiv:1911.00473* (2019).
32. Zhong, Haoxi, et al. "How does NLP benefit legal system: A summary of legall artificial intelligence." *arXiv preprint arXiv:2004.12158* (2020).
33. Williams, Christopher. *Tradition and change in legal English: Verbal constructions in prescriptive texts*. Vol. 20. Peter Lang, 2007.
34. Mellinkoff, David. *The language of the law*. Wipf and Stock Publishers, 2004.
35. Mertz, Elizabeth. *The language of law school: learning to" think like a lawyer"*. Oxford University Press, USA, 2007.
36. Haigh, Rupert. *International legal English: A Practical Introduction for Students and Professionals*. Routledge, 2020.
37. Kovaleva, Olga, et al. "Revealing the dark secrets of BERT." *arXiv preprint arXiv:1908.08593* (2019).
38. Voita, Elena, et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned." *arXiv preprint arXiv:1905.09418* (2019).
39. Strubell, Emma, Ananya Ganesh, and Andrew McCallum. "Energy and policy considerations for deep learning in NLP." *arXiv preprint arXiv:1906.02243* (2019).
40. Gong, Yunchao, et al. "Compressing deep convolutional networks using vector quantization." *arXiv preprint arXiv:1412.6115* (2014).
41. S Han, J. Pool, J. Tran, and W. Dally. 2015. Learning both weights and connections for efficient neural network. In NIPS.
42. G. Hinton, O. Vinyals, and J. Dean. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.
43. Ghosh, Sreyan, et al. "Dale: Generative data augmentation for low-resource legal nlp." *arXiv preprint arXiv:2310.15799* (2023).
44. Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." *arXiv preprint arXiv:1908.10063* (2019).
45. Lee, Jinhyuk, et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining." *Bioinformatics* 36.4 (2020): 1234-1240.

46. Rasmy, Laila, et al. "Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction." *NPJ digital medicine* 4.1 (2021): 86.
47. Habibi, Maryam, et al. "Deep learning with word embeddings improves biomedical named entity recognition." *Bioinformatics* 33.14 (2017): i37-i48.
48. Moen, S. P. F. G. H., and Tapio Salakoski2 Sophia Ananiadou. "Distributional semantics resources for biomedical text processing." *Proceedings of LBM* (2013): 39-44.
49. Yang, Yi, Mark Christopher Siy Uy, and Allen Huang. "Finbert: A pretrained language model for financial communications." *arXiv preprint arXiv:2006.08097* (2020).
50. Malo, Pekka, et al. "Good debt or bad debt: Detecting semantic orientations in economic texts." *Journal of the Association for Information Science and Technology* 65.4 (2014): 782-796.
51. Champin, Pierre-Antoine, Fabien Gandon, and Lionel Médini. *WWW'18: Companion Proceedings of the The Web Conference 2018*. ACM, 2018.
52. Cheng, Yu, et al. "A survey of model compression and acceleration for deep neural networks." *arXiv preprint arXiv:1710.09282* (2017).
53. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).