Performance-Guided LLM Knowledge Distillation for Efficient Text Classification at Scale

Flavio Di Palo* **Prateek Singhi*** **Bilal Fadlallah**
Amazon
{paloflav, snghips, bhf}@amazon.com

Abstract

Large Language Models (LLMs) face significant challenges at inference time due to their high computational demands. To address this, we present Performance-Guided Knowledge Distillation (PGKD), a cost-effective and high-throughput solution for production text classification applications. PGKD utilizes teacher-student Knowledge Distillation to distill the knowledge of LLMs into smaller, task-specific models. PGKD establishes an active learning routine between the student model and the LLM; the LLM continuously generates new training data leveraging hard-negative mining, student model validation performance, and early-stopping protocols to inform the data generation. By employing a cyclical, performance-aware approach tailored for highly multi-class, sparsely annotated datasets prevalent in industrial text classification, PGKD effectively addresses training challenges and outperforms traditional BERT-base models and other knowledge distillation methods on several multi-class classification datasets. Additionally, cost and latency benchmarking reveals that models fine-tuned with PGKD are up to 130X faster and 25X less expensive than LLMs for inference on the same classification task. While PGKD is showcased for text classification tasks, its versatile framework can be extended to any LLM distillation task, including language generation, making it a powerful tool for optimizing performance across a wide range of AI applications.

1 Introduction

Successful research in Artificial Intelligence (AI) and Large Language Models (LLMs) brought significant interest to the industrial applications of LLMs. LLMs mainly refer to Transformer-based (Vaswani et al., 2017) neural language models that contain tens to hundreds of billions of parameters, which are pre-trained on massive text data, such as Mistral 7B (Jiang et al., 2023), LLaMA (Touvron et al., 2023), and GPT-4 (OpenAI et al., 2024). Compared to smaller Pre-trained Language Models (PLMs) like BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), LLMs not only have significantly larger model sizes but also demonstrate stronger language understanding and generation capabilities. It has been demonstrated that LLMs can achieve considerable performance with limited task-specific annotated data across several domains, including natural language understanding, question answering, and code generation (Minaee et al., 2024). In the text classification domain, LLMs like GPT-3 outperforms ad-hoc state-of-the-art (SOTA) models on several benchmarks by just using few-shot prompting techniques (Sun et al., 2023).

While the utility of LLMs is evident, their application at scale is challenged by high inference latency and cost. LLMs are unsuitable for production environments where Service-Level Agreements (SLAs) are strict and scalability is essential. Furthermore, many Natural Language Processing (NLP) tasks in the industry only require a subset of the LLMs capabilities, e.g., Intent Detection, a multi-class classification problem that can be solved by fast and inexpensive PLM and does not require language generation capabilities. Furthermore, utilizing LLMs for text classification restricts the nuanced output information derived from fine-tuning a PLM. PLMs, when equipped with a classification head, produce output vectors with predicted probabilities that reflect the model's certainty about the classification results. However, this level of detail is not easily attainable from LLMs. LLMs, instead, generate plain text that requires parsing the model output and handling hallucinations, complicating production systems even further for simple tasks like text classification.

Considering the challenges presented by LLMs along with the effectiveness they bring, we propose

*Equal contribution.

a novel method called Performance-Guided Knowledge Distillation (PGKD). The proposed methodology aims to enhance the performance of smaller PLMs, which already meet inference SLA and cost constraints, by using LLMs as teachers during training.

2 Related Work

Knowledge Distillation (KD), introduced by (Hinton et al., 2015), is a promising technique to transfer the capabilities of complex and high-maintenance models to more compact and efficient student models. The core idea is to train a lean student model to mimic the soft probabilities generated by a more complex and costly teacher model.

Using LLMs for ground truth generation and as KD teacher has found success in multiple tasks (Chiang and yi Lee, 2023; Gilardi et al., 2023; Chan et al., 2023). Recent research (Gu et al., 2024) has explored novel KD with open-source LLMs and loss functions explicitly tailored for teacher-student distillation. The same work also summarizes the two commonly applied categories of KD: white-box KD (Gou et al., 2021), where the teacher parameters are available to use for model distillation, and black-box KD, where only the teacher predictions are accessible. black-box KD is less restrictive in terms of structural requirements for teacher models and student models, and can use closed-source teacher models (such as ChatGPT API); additionally, it does not require the private deployment of teacher models and does not present challenges in custom loss function convergence.

Most recent black-box KD methods mainly focus on LLM data generation or augmentation. (Lou et al., 2023) proposed AugGPT, a text data augmentation approach based on ChatGPT that rephrases each sentence in the training samples into multiple conceptually similar but semantically different samples. ZeroGen (Ye et al., 2022) is a flexible and efficient zero-shot learning method, providing insights on data-free, model-agnostic knowledge distillation. ZeroShotDataAug (Ubani et al., 2023) investigates ChatGPT-generated synthetic training data to augment low-resource datasets, outperforming existing data augmentation approaches, and explores methodologies for evaluating the quality of the generated augmented data. SunGen (Gao et al., 2022) optimizes synthetic data generation in black-box knowledge distillation by employing adversarial sampling and iterative refinement to create diverse, challenging data that aligns with the teacher model's knowledge boundaries.

Despite these advancements, the potential of LLMs in KD extends beyond mere data generation. Recent studies have explored dynamic interactions between teacher and student models at training time for effective KD. (Liu et al., 2024) proved the effectiveness of active learning for teacher-student distillation for LLMs and introduces the EvoKD framework. This framework is specifically tailored to enhance the capabilities of a student model within a few-shot learning scenario, where only a limited quantity of training data is available.

Research on text classification using KD has been focused on few-shot learning scenarios, such as 1-shot or 5-shot, where 1 to 5 training samples are provided to the base model, typically narrowing the task to binary classification. While this focus offers valuable insights for theoretical exploration, it does not adequately represent the complexity of text classification tasks encountered in industrial applications. In a typical industrial setting, classification challenges, such as intent detection, topic classification, and customer feedback analysis, involve a much larger number of categories. Moreover, in these settings, annotating hundreds to a few thousand samples is often feasible and economically viable, making the 1-shot or 5-shot evaluation unrealistic given the availability of annotated data.

3 Methods

PGKD is an iterative KD algorithm that aims to enhance the application of KD in highly multi-class, sparsely annotated datasets common in industrial environments. The primary motivation for this work is to leverage active learning to strengthen the connection between the student and teacher models in the distillation process. PGKD gives the teacher model direct and continued visibility into the learning status of the student model also making it generically extensible to wide variety of learning tasks. PGKD addresses limitations of recent works in black-box KD, in particular: (1) EvoKD only shows results on binary-classification benchmark datasets, a setting that is infrequent in practical machine learning applications; (2) ZeroGen only relies on the latent knowledge of the teacher and is not aware of the student's status; the teacher model in EvoKD is also not provided with an overview of the latest student model performance on all the available classes and only shows

the LLM a few misclassified examples; while this approach is useful in the context of binary text classification, it can be limiting in a highly multi-class context where many classes might not be represented in the sample provided to the teacher. (3) In SunGen, distillation focuses on the capability of the teacher model to generate high entropy data; EvoKD too focusses on correctly classified and misclassified samples but ignores emphasizing the information that is contained in the "Hard Negative Samples", i.e., incorrectly classified samples which the student model is confident about. (4) previous works do not provide a clear termination criterion for the iterative algorithm but treat the number of epochs for the knowledge distillation process as a hyperparameter that needs to be tuned for each dataset.

The proposed PGKD overcomes the limitation above by leveraging the following techniques.

(1) Gradual Evaluation Checks: At each KD step, the student model is evaluated on the validation set, and a report of student validation metrics like Accuracy, Precision, Recall, and F1 on each class is inserted in the teacher model prompt (Appendix A). This approach allows the teacher model to observe the student's overall performance and helps the teacher model guide the direction of the optimization. For instance, a low accuracy on a specific class will signal the teacher model to generate more relevant data samples for that class; this makes the KD process performance-aware and allows the teacher model to observe and directly optimize the student model performance. It also induces automatic class imbalance handling that is directly based on student performance. It is important to remark that in this process, only high-level validation metrics are shared with the LLM, but no validation sample is leaked. This ensures the mitigation of the crucial risk of over-fitting the validation set during the learning process.

(2) Hard Negative Mining: Hard Negative samples are defined as the misclassified samples in the training set for which the student model is more confident in its decision. These samples are generally the most informative ones, being the closest to the student model decision boundary. PGKD includes the Hard Negative samples from the previous training run in the teacher's prompt. The LLM evaluates the student model's performance deficiencies by accessing sentence patterns likely to result in errors. The confidence of the student model is computed by considering the values from the classification head of the student model. This approach allows the teacher model to gain rich insights into which examples the student is struggling to learn and encourages the teacher to produce new samples that help overcome these crucial blind spots.

(3) Early Stopping: To maximize the student model's distillation while preventing performance drift and overfitting PGKD uses early stopping to control the overall validation loss. PGKD returns the model having the lowest validation loss as the best model to be used for testing.

The PGKD methodology is detailed in Algorithm 1 and a schema of the same is reported in Figure 1 and Figure 2.

The algorithm starts by initializing and training a baseline model on an initial dataset D^0 composed of a thousand annotated samples, divided into 80% training and 20% validation. The base model obtained after this training is denoted as $model_0$. PGKD iteratively refines $model_0$ performance; during each iteration, the PGKD algorithm identifies correctly classified $D^i_{correct}$ and misclassified examples $D^i_{incorrect}$, as well as hard negatives $D^i_{hard\_negatives}$. PGKD then computes the model $val\_results$ on the validation set D^{val} and leverages an LLM to generate new training data D^{i+1} tailored to these findings according to $PGKD\_prompt$. $PGKD\_prompt$ includes dataset classification taxonomy for the specific task and a limited number of few-shot samples from the train set, prompt is reported in Appendix A. The PGKD process runs for num_kd_steps and if the validation loss does not improve consecutively beyond the $patience\_limit$, the algorithm terminates early to prevent overfitting. At the end of the PGKD process, the best model on the validation set, $model^{best}$ is returned.

4 Datasets and Experiments

Our experiments focus on four multi-class classification datasets: AG-news (Gulli, 2005), Yahoo Answers (Ardeshna, 2020), Huffington Post (Misra, 2018), and AMZN Reviews (Kashnitsky, 2018), described in Table 1. The datasets cover a wide range of number of classes, from 4 to 335. These experiments aim to study the PGKD performance at the variation of the number of classes in the dataset. The base PLM model used for PGKD is BERT-base model, as done in (Liu et al., 2024); the model has been fine-tuned with a classification head using categorical cross-entropy leverag-

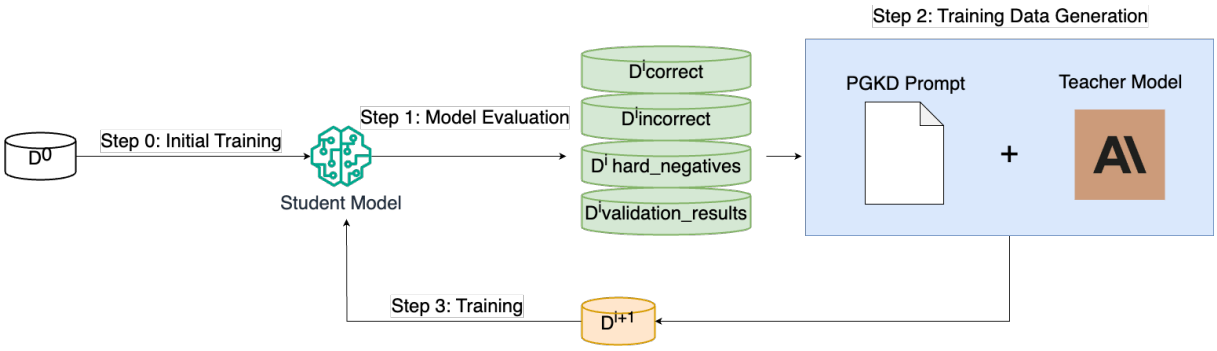
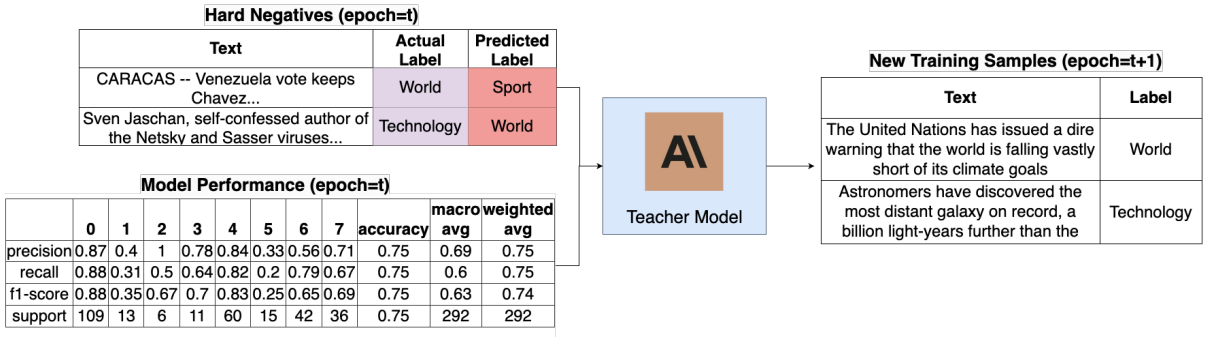


Figure 1: PGKD process, student model is initially trained on a set of labeled samples, then the KD process starts. The LLM generates new training samples for the student model based on the student's correct/misclassified samples, hard negative samples, and a report of the student's validation metrics.



Hard Negatives (epoch=t)

Text	Actual Label	Predicted Label
CARACAS -- Venezuela vote keeps Chavez...	World	Sport
Sven Jaschan, self-confessed author of the Netsky and Sasser viruses...	Technology	World

Model Performance (epoch=t)

	0	1	2	3	4	5	6	7	accuracy	macro avg	weighted avg
precision	0.87	0.4	1	0.78	0.84	0.33	0.56	0.71	0.75	0.69	0.75
recall	0.88	0.31	0.5	0.64	0.82	0.2	0.79	0.67	0.75	0.6	0.75
f1-score	0.88	0.35	0.67	0.7	0.83	0.25	0.65	0.69	0.75	0.63	0.74
support	109	13	6	11	60	15	42	36	0.75	292	292

New Training Samples (epoch=t+1)

Text	Label
The United Nations has issued a dire warning that the world is falling vastly short of its climate goals	World
Astronomers have discovered the most distant galaxy on record, a billion light-years further than the	Technology

Figure 2: PGKD methods leverages Validation Metrics and Hard Negative samples from $epoch_t$ in the LLM prompt. The teacher model generates new samples to augment the training set for $epoch_{t+1}$.

Algorithm 1 Performance-Guided Knowledge Distillation.

Require: D^0, D^{val} *num_kd_steps*, *patience_limit*, PGKD_prompt

Initialize $model$; $model^0 \leftarrow$ train $model$ on D^0; $i \leftarrow 0$; $step \leftarrow 0$
$best\_validation\_loss \leftarrow \infty$; $val\_results \leftarrow$''; $patience\_counter \leftarrow 0$; $history \leftarrow \{D^0\}$
while $i <= num\_kd\_steps$ **do**
  $D^i_{incorrect}, D^i_{correct} \leftarrow$ Find_Correctly_Classified_And_Misclassfied_Examples$(model^i, D^i)$
  $D^i_{hard\_negatives} \leftarrow$ Find_Hard_Negative_Examples$(model^i, D^i)$
  $D^{i+1} \leftarrow$ LLM$(D^i_{incorrect}, D^i_{correct}, D^i_{hard_negatives}, val_results, PGKD_prompt)$
  Add D^{i+1} to $history$
  $model^{i+1} \leftarrow train\ model\ on\ history$
  $new\_loss, val\_results \leftarrow evaluate(model^i, D^{val})$
  if $new\_loss > best\_validation\_loss$ **then**:
    $patience\_counter+ = 1$
    if $patience\_counter > patience\_limit$ **then**:
      $break$
    end if
  else
    $best\_validation\_loss \leftarrow new\_loss$
    $model^{best} \leftarrow model^i$
  end if
  $i \leftarrow i + 1$
end while
return $model^{best}$

Dataset	Description	#Training Samples	#Testing Samples	#Classes
AG-News	Categorize news articles gathered from more than 2000 news sources in four categories (World, Sports, Business, Sci/Tech).	120,000	7,600	4
Yahoo Answers	Categorizes answers from the popular website 'Yahoo Answers' into ten different categories.	1,400,000	60,000	10
Huffington Post	Categorizes Huffington Post news headlines from 2012–2022 into forty-one different categories such as politics, wellness, entertainment, and travel.	160,000	40,000	41
AMZN Reviews	Identifies Amazon Product categories based on the review title and text for that product. "level 3" classes present in the training sample used are considered targets for the classification task.	40,000	10,000	335

Table 1: Detailed description of the datasets used for the PGKD experiments. Each dataset varies significantly in the number of classes, training samples, and testing samples, reflecting a broad range of classification challenges.

ing the Hugging Face library[1]. A sample of 1000 annotated data points is used for $model\_0$ training and validation (80% training, 20% validation). The teacher model for the experiments is Claude-3 Sonnet accessed via AWS Bedrock[2]. Early stopping is applied on the validation loss for the initial BERT-base model and the PGKD methodology is then applied to the resulting model. For robustness of the provided results, 5 different training and validation sets of 1000 samples are chosen; results are averaged across these 5 samples. For BERT-base $model\_0$ fine-tuning, the following parameters have been set: maximum sequence length of 512, batch size of 64, fine-tuned the model for 30 epochs with a learning rate of 2×10^{-5}, and patience parameter for early stopping at 5. PGKD fine-tuning is tasked to produce 32 new samples at each iteration and the prompt takes as input 16 samples from the training set as few-shot samples used to guide the LLM data generation; the number of correct samples, incorrect samples, and hard negative samples is set to 16. The number of PGKD epochs is set to 10 and PGKD patience is set to 5. The purpose of the experiments is to measure the performance lift registered by applying the PGKD routine on top of the $model\_0$ BERT-base model for different datasets with a varying number of classes.

5 Results and Discussion

PGKD results are presented in Table 2. PGKD exhibits a varying degree of effectiveness across the datasets. **AG-news (4 classes)**: The PGKD implementation yields only slight improvements in AG-news, a dataset with a reduced number of classes. The metrics show an increase in Accuracy from 0.884 to 0.895, Macro Average F1 from 0.884 to 0.894, and Weighted Average F1 from 0.884 to 0.894 compared to BERT-base. These modest gains likely reflect the already high baseline performance, which limits the scope for further significant enhancements through the PGKD method. **Yahoo Answers (10 classes)**: PGKD shows a noticeable improvement in all measured metrics. Accuracy is enhanced from 0.649 to 0.685, Macro Average F1 from 0.657 to 0.688, and Weighted Average F1 from 0.657 to 0.688. This dataset's moderate number of classes indicates that PGKD is particularly effective in scenarios with intermediate complexity, utilizing the teacher model's strengths more effectively. **Huffington Post (41 classes)**: The application of PGKD in the Huffington Post dataset, with a substantially higher number of classes, also leads to significant improvements. Accuracy improves

[1]https://huggingface.co/docs/transformers/model_doc/bert
[2]https://aws.amazon.com/bedrock/

	Method	Accuracy	Macro Avg. F1	Weighted Avg. F1
AG-news (#classes = 4)	BERT-base	0.884 ± 0.012	0.884 ± 0.013	0.884 ± 0.011
	BERT-base + PGKD	**0.895 ± 0.014**	**0.894 ± 0.018**	**0.894 ± 0.015**
	Claude-3 (Zero-Shot)	0.826 ± 0.020	0.821 ± 0.016	0.822 ± 0.012
	SOTA - Full shot (Yang et al., 2019)	0.955	-	-
Yahoo Answers (#classes = 10)	BERT-base	0.649 ± 0.013	0.657 ± 0.017	0.657 ± 0.015
	BERT-base + PGKD	**0.685 ± 0.015**	**0.688 ± 0.019**	**0.688 ± 0.011**
	Claude-3 (Zero-Shot)	0.680 ± 0.016	0.676 ± 0.018	0.676 ± 0.017
	SOTA - Full Shot (Sun et al., 2019)	0.776	-	-
Huffington Post (#classes = 41)	BERT-base	0.474 ± 0.011	0.214 ± 0.014	0.411 ± 0.013
	BERT-base + PGKD	**0.519 ± 0.012**	0.330 ± 0.018	**0.495 ± 0.020**
	Claude-3 (Zero-Shot)	0.442 ± 0.015	**0.332 ± 0.019**	0.435 ± 0.016
	SOTA - 5-way, 5-shot (Chen et al., 2022)	0.653	-	-
AMZN Reviews (#classes = 335)	BERT-base	0.320 ± 0.014	0.074 ± 0.019	0.244 ± 0.011
	BERT-base + PGKD	**0.443 ± 0.017**	0.159 ± 0.012	0.382 ± 0.014
	Claude-3 (Zero-Shot)	0.416 ± 0.012	**0.364 ± 0.016**	**0.414 ± 0.013**
	SOTA - 5-shot (Zhang et al., 2022)	0.495	-	-

Table 2: Comparison of average and standard deviation for Accuracy, Macro Average F1, and Weighted Average F1 scores for BERT-base and BERT-base enhanced with PGKD across various datasets with differing numbers of classes. Results are shown alongside Claude-3 zero-shot and SOTA full-shot or few-shot methods as referenced in the literature.

from 0.474 to 0.519, Macro Average F1 from 0.214 to 0.330, and Weighted Average F1 from 0.411 to 0.495. These improvements underscore the benefits of PGKD in managing more complex class structures and enhancing model performance. **AMZN Reviews (335 classes)**: This dataset, featuring the largest number of classes, displays the most dramatic improvements with PGKD. Accuracy significantly increases from 0.320 to 0.443, Macro Average F1 from 0.074 to 0.159, and Weighted Average F1 from 0.244 to 0.382. These results highlight PGKD's strong capability in handling extensive, complex class structures, particularly beneficial in contexts with scarce labeled data. We observe a correlation between the number of classes and the improvement margin achieved through PGKD. Datasets with a lower number of classes show negligible improvements, while datasets with a larger number of classes show significant performance gains. This suggests that the methodology is particularly suited for production applications where distinguishing between a large number of classes is challenging due to the complexity and sparsity of the annotated data; this class of problems is prevalent in many industrial Machine Learning (ML) applications. Performance of the current SOTA methodology for each dataset and performance of Claude-3 Sonnet zero-shot classification are reported. The corresponding prompt for zero-shot classification is reported in Appendix A. In all the reported use-cases, PGKD outperformed zero-shot Claude-3 Accuracy improving the performance gap between BERT-base and current SOTA; BERT-base + PGKD outperforms Claude-3 zero-shot on Weighted Average F1 on all datasets except AMZN Reviews; BERT-base + PGKD has superior Macro Average F1 results when compared with Claude-3 on two out of four datasets, with slightly lower results on the Huffington Post dataset. It is plausible that enhancing the prompt with complex few-shot techniques could elevate the LLM to match the current SOTA performance, as reported in (Sun et al., 2023); however, further LLM prompt exploration is beyond the scope of this work. While PGKD is expected to be maximally useful when the number of annotated data is limited, it is interesting to evaluate performance gains obtained by PGKD as the number of training samples increases. Figure 3 presents the results for the AMZN Reviews dataset obtained at a varying number of training dataset sizes, with similar trends observed across all datasets; full results are documented in Appendix B. The PGKD approach consistently surpassed the performance of the BERT-base model. Notably, as the number of training samples from the original dataset increases, the performance differential between PGKD and the original model decreases. This phenomenon can be attributed to

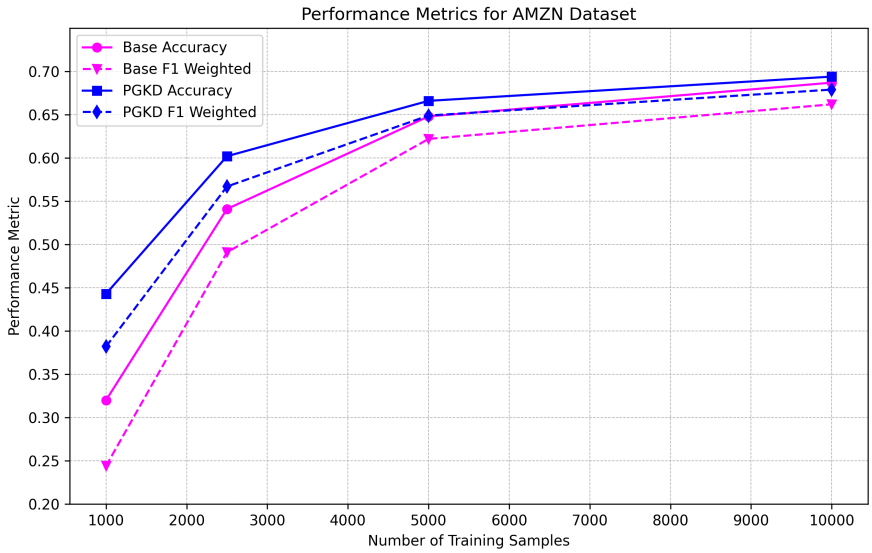


Figure 3: Performance of PGKD across varying training set sizes on the AMZN Reviews dataset.

the diminishing returns of new data generation by the LLM as the volume of original dataset samples grows. Interestingly, while PGKD performance improvement reduces with the increasing number of training samples, we can observe that the PGKD provides the best results across all dataset considered for all training sample sizes. PGKD demonstrates to not degrade model in any of the experiments conducted.

5.1 Comparative Analysis of Related Literature

We compared PGKD with other KD strategies on two datasets: the IMDB Dataset (Pathi, 2018), containing 2 classes, and the Inshorts News V7 (Chander, 2021), containing 7 classes, as presented in (Liu et al., 2024). Table 3 reports the comparison between PGKD, EvoKD and other baseline 1-shot classification methods from (Liu et al., 2024). The 1-shot approach discussed in (Liu et al., 2024) may be impractical for production environments, where access to larger datasets is the norm. The BERT-base model trained with 1000 samples achieves a weighted average F1 score of 0.872 on the IMDB dataset and 0.933 on the Inshorts dataset. Interestingly BERT-base already outperforms all the 1-shot methodologies reported in (Liu et al., 2024) even without PGKD. PGKD further improves upon this performance, achieving a weighted average F1 score of 0.908 on the IMDB dataset and 0.943 on the Inshorts dataset. The results demonstrate that PGKD is able to improve BERT-base model performance in this context as well.

Method	**IMDB**	**Inshorts**
Full Shot	0.949	0.970
BERT-base (n=1000)	0.872	0.933
BERT-base + PGKD (n=1000)	**0.908**	**0.943**
No Augment (1-shot)	0.583	0.640
EDA (1-shot)	0.618	0.677
ZeroGen (1-shot)	0.508	0.833
SunGen (1-shot)	0.576	0.810
Gradual (1-shot)	0.685	0.760
AugGPT (1-shot)	0.690	0.790
EvoKD (1-shot)	0.798	0.851
EvoKD + Init (1-shot)	0.835	0.868

Table 3: Weighted Average F1 score comparison of Knowledge Distillation methods on English IMDB and Inshorts datasets. 1-shot methods reported from (Liu et al., 2024).

5.2 Ablation Study

Table 4 reports the effectiveness of specific PGKD components, proving the contribution of the proposed methodology over a general LLM active learning framework. Experiments have been carried out with the PGKD methodology by selectively disabling the Validation Report and Hard Negatives features across different datasets. The removal of the Validation component (w/o Validation), which uses performance metrics to guide the distillation

Method	AG-news	Yahoo A.	Huffington	AMZN
PGKD	0.895	0.685	0.519	0.443
w/o Validation	0.893	0.669	0.501	0.419
w/o Hard Negatives	0.887	0.675	0.510	0.433

Table 4: Average Accuracy on five training samples; PGKD applied to BERT-base model trained with 1000 samples.

process, resulted in a decrease in accuracy across all datasets. Specifically, the accuracy decreased from 0.895 to 0.893 on AG-news, from 0.685 to 0.669 on Yahoo Answers, from 0.519 to 0.501 on Huffington Post, and more significantly from 0.443 to 0.419 on Amazon Reviews. This indicates that validation metrics information is useful in the distillation process by making it performance-aware; as expected, this feature is particularly relevant to datasets with a high number of classes. Removing Hard Negative Mining (w/o Hard Negatives), which incorporates challenging misclassified samples to refine the student model's decision boundaries, led to a more subtle drop in model performance. For instance, accuracy fell from 0.895 to 0.887 on AG-news, from 0.685 to 0.675 on Yahoo Answers, from 0.519 to 0.510 on Huffington Post, and from 0.443 to 0.433 on Amazon Reviews. This decline was more pronounced in datasets with complex classification tasks, highlighting the importance of hard negative samples in enhancing the robustness and accuracy of the student model in multi-class scenarios.

5.3 Cost and Latency Benchmarking

Inference latency and cost for PGKD and LLMs[3] are reported in Table 5. BERT-base + PGKD processes a batch of 64 inference samples in 21.45 seconds on an AWS m5.4xlarge instance (16 vCPUs) for $0.0046, and in 0.46 seconds on a g5.4xlarge GPU instance for $0.01. Claude Sonnet inference takes 60.64 seconds per batch on average, costing $0.38, for inputs averaging 1k tokens. Cheaper open-source models like LLaMA 3 8B take 58.06 seconds on average and cost $0.06 per batch. LLMs inference for the proposed classification task is approximately 3X slower and 25X more expensive than BERT-base + PGKD on a CPU instance and approximately 130X slower and 6X more expensive on a GPU instance.

Method	Latency (s)	Cost ($)
BERT-base + PGKD (CPU)	21.45	0.0046
BERT-base + PGKD (GPU)	0.46	0.0107
Claude-3 zero-shot	60.64	0.3867
LLaMA 3 8B zero-shot	58.05	0.0609

Table 5: Comparison of latency in seconds and cost in dollars for BERT-base models enhanced with PGKD (both CPU and GPU configurations), Claude-3 zero-shot, and LLaMA 3 8B zero-shot models.

[3]All LLMs are accessed via Amazon Bedrock: https://aws.amazon.com/bedrock/

6 Conclusion and Future Work

Deploying LLMs in real-world text classification applications poses significant challenges due to high inference costs and latency. This research introduces PGKD, a novel methodology designed to effectively distill LLM knowledge into faster, more efficient models for multi-class text classification. This study proves the substantial performance improvements of PGKD distillation compared to regular fine-tuning of a BERT-base PLM, while maintaining limited inference latency and costs. Comparative analyses with existing knowledge distillation and augmentation strategies further underscore PGKD's practical performance enhancements. The proposed ablation study reveals the significant contributions of PGKD components, such as Gradual Performance Checks on Validation Reports and Hard Negative Mining. Cost and latency benchmarks provide compelling evidence of PGKD's efficiency. Compared to zero-shot LLM costs, BERT-base with PGKD has proven to be significantly more cost-effective and up to 130X faster for a broad spectrum of multi-class classification tasks. Future research will investigate the impact of different teacher LLMs on the distillation process and explore the influence of the student model size on PGKD effectiveness. Future research will also focus on exploring more advanced prompting techniques to optimize PGKD performance. While PGKD is showcased for text classification tasks, its versatile framework can be extended to any LLM distillation task, including language generation, making it a powerful tool for optimizing performance across a wide range of AI applications.

7 Limitations

While PGKD has demonstrated significant improvements in multi-class text classification tasks, there are some limitations to consider the following aspects.

(1) Dependence on LLM performance: The effectiveness of PGKD is inherently tied to the performance of the LLM used for knowledge distillation. If the LLM struggles with domain-specific data or fails to generate high-quality samples, it may limit the potential gains from the distillation process. Future work could explore the impact of using different LLMs or ensembles of LLMs to mitigate this limitation.

(2) Computational cost during distillation: Although PGKD results in a more efficient student model for inference, the distillation process itself can be computationally expensive due to the iterative generation of samples from the LLM. This may limit the scalability of the approach for vast datasets or frequent model updates.

(3) Evaluation on a limited set of tasks: While PGKD has been evaluated on diverse datasets, the experiments are still limited to a specific set of multi-class text classification tasks. Further validation in a broader range of datasets, languages, and domains would strengthen the generalizability of the findings. Additionally, exploring the applicability of PGKD to other NLP tasks beyond classification, such as named entity recognition or question answering, could broaden its impact. This is something that will be addressed in future work.

(4) Sensitivity to prompt engineering: The performance of PGKD may be sensitive to the quality of the prompts used to guide the LLM in generating samples and poorly designed prompts could lead to suboptimal distillation results. Developing robust prompt engineering strategies or automating the prompt generation process could help mitigate this limitation and improve the consistency of PGKD's performance across different datasets and tasks.

Addressing these limitations could further enhance the applicability and robustness of the PGKD methodology, making it an even more valuable tool for leveraging LLMs distillation in production-ready systems.

8 Ethical Considerations

8.1 Benefits

The proposed PGKD methodology has significant societal implications, particularly in democratizing access to high-performance models in cost-constrained applications. By leveraging the knowledge distilled from LLM, PGKD enables the development of more accurate and efficient models that can be deployed in a wide range of applications, including intent detection, topic classification, and other multi-class text classification tasks.

In many industrial and commercial settings, the high inference cost and latency of LLM can be a significant barrier to adoption. PGKD offers a cost-effective and efficient solution, enabling organizations to leverage the power of LLMs without the associated costs and latency at inference time. This can profoundly impact the democratization of access to AI technology, particularly in resource-constrained environments.

Furthermore, PGKD has the potential to benefit a wide range of industries and applications, including customer support, messaging platforms, and other areas where multi-class text classification is a critical component. By enabling the development of more accurate and efficient models, PGKD can help to improve the overall quality of service and user experience in these applications.

8.2 Potential Risks

The development and application of PGKD for multi-class text classification present potential fairness and bias considerations as PGKD's performance and the quality of student model outputs depends heavily on LLM output quality. If the teacher model for PGKD contains bias or even worse hallucinates, it will generate biased and even hypothetical data points on which the student model will be trained. The distilled LLM knowledge may perpetuate and amplify existing student model biases and even impact training data quality. Ensuring the use of LLMs that are robust against bias and are well grounded in data generation is essential in PGKD distillation.

References

Bhavik Ardeshna. 2020. Yahoo email classification. https://www.kaggle.com/datasets/bhavikardeshna/yahoo-email-classification. Retrieved June 15, 2024.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *Preprint*, arXiv:2308.07201.

Shashi Chander. 2021. Inshorts news data. https://www.kaggle.com/datasets/shashichander009/inshorts-news-data. Retrieved June 15, 2024.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2022. Contrastnet: A contrastive learning framework for few-shot text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:10492–10500.

Cheng-Han Chiang and Hung yi Lee. 2023. Can large language models be an alternative to human evaluations? *Preprint*, arXiv:2305.01937.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2022. Self-guided noise-free data generation for efficient zero-shot learning. In *International Conference on Learning Representations*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. *Preprint*, arXiv:2306.08543.

Antonio Gulli. 2005. The anatomy of a news search engine. In *The anatomy of a news search engine*, pages 880–881.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Yury Kashnitsky. 2018. Hierarchical text classification. https://www.kaggle.com/datasets/kashnitsky/hierarchical-text-classification. Retrieved June 15, 2024.

Chengyuan Liu, Yangyang Kang, Fubang Zhao, Kun Kuang, Zhuoren Jiang, Changlong Sun, and Fei Wu. 2024. Evolving knowledge distillation with large language models and active learning. *Preprint*, arXiv:2403.06414.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal information extraction as unified semantic matching. *Preprint*, arXiv:2301.03282.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Rishabh Misra. 2018. News category dataset. https://www.kaggle.com/datasets/rmisra/news-category-dataset. Retrieved June 15, 2024.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li,

Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Lakshmi Pathi. 2018. Imdb dataset of 50k movie reviews. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. Retrieved June 15, 2024.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Language Models are Unsupervised Multitask Learners*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, pages 194–206. Springer.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *Preprint*, arXiv:2304.14334.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *Preprint*, arXiv:2202.07922.

Jianhai Zhang, Mieradilijiang Maimaiti, Gao Xing, Yuanhang Zheng, and Ji Zhang. 2022. MGIMN: Multi-grained interactive matching network for few-shot text classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1937–1946, Seattle, United States. Association for Computational Linguistics.

A Prompt Details

This section reports the prompts used in this work: 'PGKD Prompt' is the prompt used for the Knowledge Distillation process.'Zero-shot Classification Prompt' is a prompt used for zero-shot classification with LLM.

A.1 PGKD Prompt

Human:
You are a Teacher model for a Student LM to perform topic detection on the following taxonomy:
{**Dataset Class Taxonomy**}
Here are a few labeled examples that show the correct label for this task:
{**Few-Shot Labeled Samples from Training Dataset**}
Given the current model performance, please generate {**PGKD Batch Size**} training samples for the model to improve its performance. The response should be a list of dictionaries in JSON format, the response needs to be parsable so do not output anything else rather than the response itself. The objective is to maximize the model accuracy, generate new samples knowing that the classification report over validation set is:
{**Classification Report on the Validation Set**}
Please consider a few samples that the model was able to classify correctly:
{**Correctly Classified Samples with correct label and Student-predicted label**}
And samples the model was not able to classify correctly:
{**Misclassified Samples with correct label and Student-predicted label**}
The model has a high confidence in classifying the following misclassified examples:
{**Hard Negative Samples with correct label and Student-predicted label**}
Assistant:

A.2 Zero-shot Classification Prompt

Human:
You are an AI assistant, and you are tasked to perform topic classification starting from text. You are asked to classify text in topics categories. You are only allowed to choose one of the following categories:
{**Dataset Class Taxonomy**}
Please provide only one category for each text in JSON format. For example:
"class_label": , "class_names": ""
Please do not repeat or return the content back again, just provide the category in the defined format.
Text-to-classifiy:
{**Paste text for zero-shot classification**}
Assistant:

B Impact of Training Sample Size

Table 6 illustrates the performance metrics of BERT-base and BERT-base augmented with PGKD across different datasets as training samples increase. Noticeably, the improvement gap between PGKD and BERT-base narrows with more training data. For instance, in the AG-news dataset, the performance difference in accuracy is 1.1% with 1000 samples but decreases to just 0.3% with 10,000 samples. This trend is consistently observed across all datasets, highlighting that while PGKD enhances performance, its relative benefit diminishes as more training data is used. Importantly, PGKD shows no signs of performance degradation, maintaining or improving upon the baseline metrics across all sample sizes and datasets.

Dataset	**Method**					
	BERT base			BERT base + PGKD		
Samples	Acc.	Macro Avg. F1	Weighted Avg. F1	Acc.	Macro Avg. F1	Weighted Avg. F1
AG-news (#class = 4)						
1000	0.884	0.884	0.884	0.895	0.894	0.894
2500	0.899	0.899	0.899	0.910	0.909	0.909
5000	0.903	0.903	0.903	0.911	0.911	0.911
10000	0.914	0.914	0.914	0.917	0.917	0.917
Yahoo Answers (#class = 10)						
1000	0.649	0.657	0.657	0.685	0.688	0.688
2500	0.700	0.693	0.693	0.708	0.705	0.704
5000	0.702	0.697	0.697	0.719	0.707	0.709
10000	0.721	0.716	0.716	0.725	0.718	0.721
Huffington Post (#class = 41)						
1000	0.474	0.214	0.411	0.519	0.330	0.495
2500	0.537	0.343	0.519	0.553	0.394	0.541
5000	0.551	0.378	0.547	0.564	0.425	0.564
10000	0.601	0.387	0.561	0.605	0.476	0.604
AMZN Reviews (#class = 335)						
1000	0.320	0.074	0.244	0.443	0.159	0.382
2500	0.541	0.243	0.491	0.602	0.340	0.567
5000	0.648	0.404	0.622	0.666	0.436	0.649
10000	0.687	0.451	0.662	0.694	0.476	0.679

Table 6: Comparative Performance of BERT-base and BERT-base + PGKD Across Different Datasets at Increasing Training Sample Sizes.