Logits Reranking via Semantic Labels for Hard Samples in Text Classification

Peijie Huang[†], **Junbao Huang**[†], **Yuhong Xu**[*], **Weizhen Li, Xisheng Xiao**
College of Mathematics and Informatics, South China Agricultural University, China
pjhuang@scau.edu.cn, gumbouh@stu.scau.edu.cn, xuyuhong@scau.edu.cn,
leeweizhen@stu.scau.edu.cn, xishengxiao.mail@gmail.com

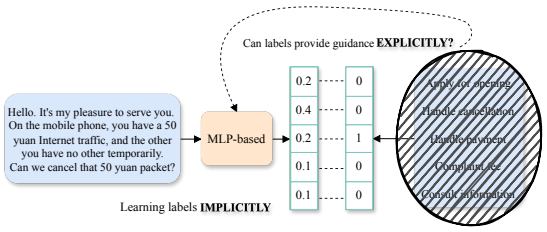
Abstract

Pre-trained Language Models (PLMs) have achieved significant success in text classification. However, they still face challenges with hard samples, which refer to instances where the model exhibits diminished confidence in distinguishing new samples. Existing research has addressed related issues, but often overlooks the semantic information inherent in the labels, treating them merely as one-hot vectors. In this paper, we propose **L**ogits **R**eranking via **S**emantic **L**abels (LRSL), a model-agnostic post-processing method that leverages label semantics and auto detection of hard samples to improve classification accuracy. LRSL automatically identifies hard samples, which are then jointly processed by MLP-based and Similarity-based approaches. Applied only during inference, LRSL operates solely on classification logits, reranking them based on semantic similarities without interfering with the model's training process. The experiments demonstrate the effectiveness of our method, showing significant improvements across different PLMs. Our codes are publicly available at https://github.com/SIGSDSscau/LRSL.

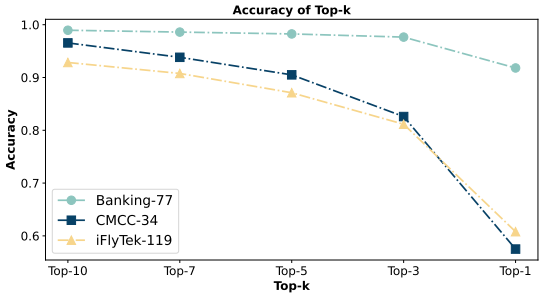
1 Introduction

Text classification tasks (Fields et al., 2024) are traditional and crucial tasks in natural language processing (NLP), holding significant importance in both academia and industry.

The Multilayer Perceptron (MLP) -based methods are conventional methods that have been widely adopted. Before the rise of pre-trained language models (PLMs), these methods have been predominantly applied in architectures such as TextCNN (Kim, 2014), HAN (Yang et al., 2016). Even after the emergence of PLMs, the MLP-based approach remains prevalent. For instance, autoencoding models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) learn bidirectional context encoders, and subsequently, an MLP is used for downstream classification tasks. Similarly, in the era of Large Language Models (LLMs), autoregressive models such as GPT-2 (Radford et al., 2019) and Llama (Touvron et al., 2023) utilize the representation of the final token in the input sequence for classification through an MLP layer. Specifically, the core idea of these methods revolves around transforming textual data into numerical vectors and mapping them through MLP layers to capture the intricate relationships between text vectors and labels. However, MLP-based methods still face challenges, particularly with diminished confidence in distinguishing new samples, referred to as hard samples, possibly due to noise in the text or



(a) MLP-based models use one-hot encoded labels for implicit learning during training. They can't truly understand the semantics of labels through one-hot encoding.



(b) Average accuracy of MLP-based PLMs from Top-10 to Top-1. CMCC-34 and iFlyTek-119 belong to long-tailed datasets, while Banking-77 does not.

Figure 1: The MLP-based models struggle to handle hard samples when learning labels implicitly.

[†] Equal contribution.
[*] Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 11250–11262

long-tail distributions in the datasets. Under such conditions, the model encounters difficulties, leading to a decline in accuracy. This shortfall arises because MLP-based methods overlook the semantic information inherent in the labels. They encode labels as one-hot vectors, which abstractly represent labels and compel the model to implicitly learn the relationship between the text and these abstract labels. As depicted in Figure 1(a), one-hot encoding represents all incorrect labels as zero, ignoring the potential relationships between labels.

Recent studies have started to consider label semantics in classification, giving rise to Similarity-based classification methods. Unlike traditional direct classification methods, they do not directly classify text but rely on the rich semantic embedding provided by PLMs for classification. For instance, Vulić et al. (2021) captured the intrinsic relationships and features among texts through prototype learning, and Mueller et al. (2022) incorporated label semantics into pre-trained generative models. However, these approaches only use similarity for classification, overlooking the fact that MLP is significantly effective for classifying easy samples.

In this paper, we combine the strengths of both methods and propose a model-agnostic post-processing approach called Logits Reranking via Semantic Labels (LRSL). This plug-and-play method enhances inference by performing logits reranking specifically on hard samples, thereby improving classification accuracy while optimizing resource consumption. We introduce a mechanism that allows the model to automatically identify hard samples, enabling the MLP-based approach to handle easy samples while the hard samples are jointly processed by the MLP-based and similarity-based approaches. Our experiments demonstrate the universality and effectiveness of our method across seven different PLMs on three challenging datasets.

Our contributions are as follows:

- We propose an efficient model-agnostic post-processing method, which only operates at the logits to rerank based on semantic similarities.
- We introduce an auto detection mechanism that lets models detect hard samples automatically.
- We conduct experiments on three challenging classification datasets with seven different models to demonstrate the universality and effectiveness of our method.

2 Related Work

2.1 Label Semantics

Label semantics has been utilized in various settings and tasks to enhance performance and robustness, even before dense embedding representations became popular. Chang et al. (2008) achieved over 80% accuracy on binary text classification tasks without any labeled training examples. Song and Roth (2014) employed a dataless approach for hierarchical and multinomial classification, demonstrating that such methods could approach or even surpass the performance of supervised approaches.

More recently, label semantics based on dense embedding representations have become widespread, especially with the rise of contextualized word embeddings (Peters et al., 2018). Gaonkar et al. (2020) utilized label embeddings derived from BERT along with a label attention mechanism to enhance emotion classification accuracy. Vulić et al. (2021) took the analogy of 'intent' being a latent semantic label where sentences associated with the intent are diverse surface instances of the class. Mueller et al. (2022) incorporated label semantics into generative models during pre-training, which explicitly injects semantic information into the model.

Our method processes hard samples by specifically targeting the reranking of logits using label semantics, which is a novel application of label semantics aimed at improving classification accuracy.

2.2 Text Embedding

Embedding models are a critical application in NLP. They encode the textual data in the latent space, where the underlying semantics of the data can be expressed by the output embeddings (Reimers and Gurevych, 2019; Ni et al., 2022).

Early approaches like one-hot encoding represented words as high-dimensional sparse vectors, which failed to capture semantic relationships between words and led to inefficiencies in computational resources. To address these issues, distributed representations such as word2vec were introduced, embedding words into dense, low-dimensional vector spaces that capture semantic similarities (Mikolov et al., 2013). ELMo generated context-sensitive embeddings using bi-directional LSTM networks, allowing word representations to change based on their context (Peters et al., 2018). Sentence-BERT (SBERT) optimized

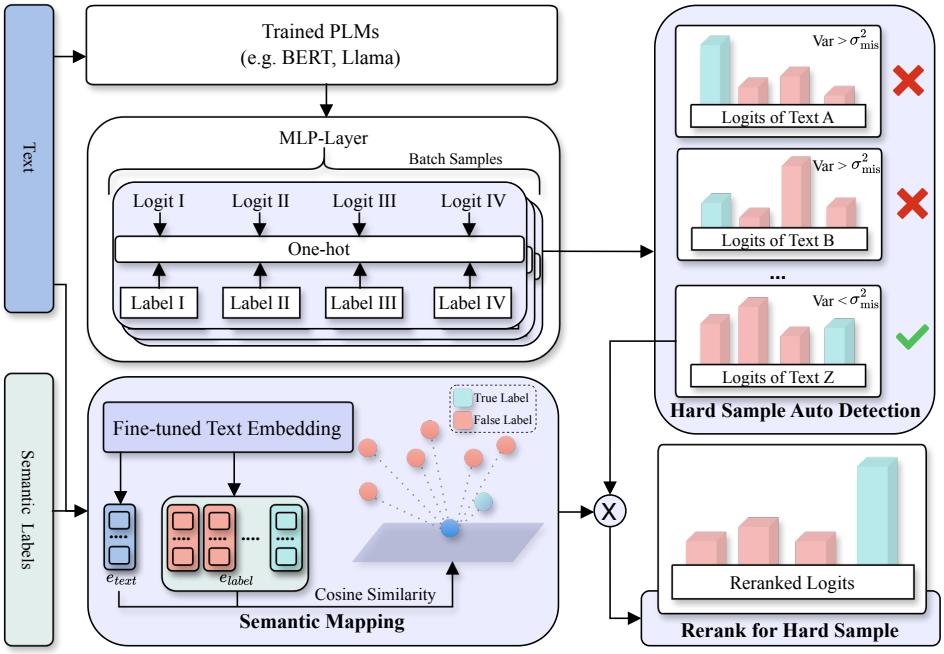


Figure 2: Overview of LRSL, which operates during the inference stage. The white sector can be replaced with any MLP-based PLMs, such as BERT, RoBERTa, or Llama. The purple sector represents the area where our approach operates. The first step is detecting hard samples automatically. Subsequently, for each hard sample, we calculate its semantic distance with all labels using fine-tuned Text Embedding. The reranking of classification logits is guided by the semantic distance.

BERT for sentence-level embeddings, enhancing its utility in tasks such as sentence similarity and information retrieval (Reimers and Gurevych, 2019).

With the advent of LLMs, the quality of text embeddings has been substantially improved, making them imperative components in information retrieval (IR). So far, there have been many impactful methods in this direction, like Contriever (Izacard et al., 2022), BGE (Xiao et al., 2023), which significantly advance the usage of text embeddings for general tasks.

We are inspired by the concept of using text embeddings in IR tasks, and transfer it to classification tasks. This allows us to leverage their rich semantic information to enhance the performance of label semantics.

3 Methodology

MLP-based PLMs have achieved remarkable success in text classification, but the effect is greatly reduced when facing hard samples. As shown in Figure 1(b), compared with the Banking-77, the model's classification accuracy tends to decrease more significantly for the other two long-tailed datasets, both of which have more hard samples, as the range of Top-k accuracy is narrowed. Transitioning from Top-3 to Top-1 reveals a sharp decline in accuracy, indicating that MLP-based models struggle to handle hard samples effectively.

Thus, we let the model detect hard samples automatically and post-process these samples. Utilizing an additional Text Embedding, we use its semantic distance between each sample and labels as the weights to rerank the logits of hard samples to enhance classification capabilities. This approach presents three challenges: How to make good use of Text Embedding? How to detect the hard samples? How to rerank the logits?

The overview of our approach is shown in Figure 2. We fine-tune an additional Text Embedding with triplets for semantic mapping (§3.1), allowing Text Embedding to provide a rich and accurate semantic space where texts and labels can be directly associated. We then propose an auto detection mechanism (§3.2), enabling the model to identify its own hard samples. For these hard samples, we use semantic distance as an alternative view of the logits, combining the two for reranking (§3.3).

3.1 Semantic Mapping

Construct Text Embedding. Following Reimers and Gurevych (2019), we use a PLM with general semantic information, such as RoBERTa, to con-

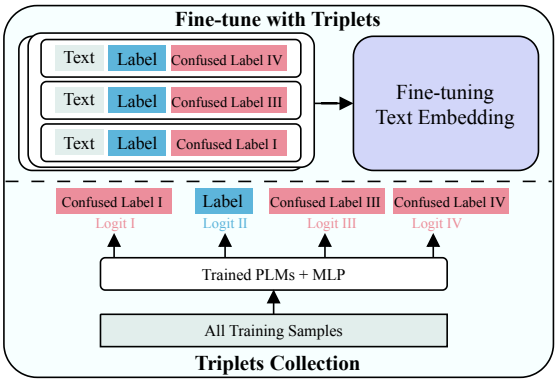


Figure 3: Fine-tuning Text Embedding with triplets. We convert the entire training samples into triplets.

struct a general Text Embedding, which provides a general semantic space so that we can map samples and labels into it. The Text Embedding model is depicted in Figure 3.

Fine-tune with Triplets. General Text Embedding is not sufficient to distinguish hard samples, so we construct triplets for Text Embedding to fine-tune. Triplets consist of text *t*, true semantic label *p*, and Top-k confused semantic labels n_k. n_k can be obtained in the MLP layer. Given the text *t*, we can obtain the hidden representation H using a PLM. H is then processed through an MLP layer to obtain the logits:

$$\boldsymbol{H} = \text{PLM}(t). \tag{1}$$

By removing the logits corresponding to the true label, we derive the logits for the confused labels. Finally, we select the indices of the Top-k most confusing labels. The formulas are as follows:

$$\boldsymbol{L} = max(\text{MLP}(\boldsymbol{H}) - \text{onehot}(p) \cdot \infty, 0), \tag{2}$$

$$n_k = \text{argsort}(\boldsymbol{L})[:k], \tag{3}$$

where *L* represents the logits of the confused labels, $\text{onehot}(p)$ represents the one-hot encoded vector of the true label p, and $[:k]$ refers to the operator that selects the Top-k values from a vector.

We select Multiple Negatives Ranking Loss (MNRL) as our fine-tuning loss function. MNRL aims to maximize the similarity between the query and the positive sample while minimizing the similarities between the query and the negative samples. In our method, we use the text as the query, the true semantic label as the positive sample, and the confused labels as the negative samples. Given the text *t* and the true semantic label *p*, we aim to maximize the similarity between t and p while minimizing the similarity between t and the Top-k confused labels $\{n_i\}_{i=1}^{k}$. The formulas are as follows:

$$\mathcal{L}_{\text{Pos}} = \exp(\text{Sim}(\text{TE}(t), \text{TE}(p))), \tag{4}$$

$$\mathcal{L}_{\text{Neg}} = \sum_{i=1}^{k} \exp(\text{Sim}(\text{TE}(t), \text{TE}(n_i))), \tag{5}$$

$$\mathcal{L}_{\text{MNRL}} = -\log \frac{\mathcal{L}_{\text{Pos}}}{\mathcal{L}_{\text{Pos}} + \mathcal{L}_{\text{Neg}}}, \tag{6}$$

where Sim denotes the cosine similarity (Salton et al., 1975). TE represents the embedding of a sample or label. $\mathcal{L}_{\text{Pos}}$ is the positive term, calculated as the exponential of the similarity between the text t and the true label p. $\mathcal{L}_{\text{Neg}}$ is the negative term, calculated as the sum of the exponentials of the similarities between the text t and the Top-k confusing labels $\{n_i\}_{i=1}^{k}$.

3.2 Hard Sample Auto Detection

Distribution Variance for Indicating Hard Samples. Even after sufficient training, the model may still struggle to distinctly differentiate the features of certain samples. This results in a uniform distribution of classification logits for these corresponding samples, indicating the model's lack of confidence in assigning appropriate labels. To address this issue, we use the variance of the Top-k distribution of logits as a metric for evaluating hard samples. Similar to Ma et al. (2023), which adopts a similar approach to identifying hard samples, our work coincides in this aspect.

In our research, evaluating model classification confidence through logits variance is essentially aligned with the concept of entropy (Shannon, 1948). Specifically, low variance in the logits indicates that the model struggles to differentiate between classes, leading to a more uniform distribution, which aligns with high entropy and represents greater uncertainty. Therefore, using logits variance as a metric is grounded in the relationship between variance, confidence, and entropy.

Auto Detection. Different models face different hard samples when dealing with various datasets. We advocate for allowing the model to autonomously identify its own hard samples rather than manually determining them. We utilize the average variance of the Top-k logits of samples that the model misclassifies σ^2_{mis} as an indicator for identifying hard samples:

$$\sigma^2_{\text{mis}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \text{Var}(\text{Top-}k(\mathbf{z}_i)), \tag{7}$$

where $\mathcal{M}$ is the set of misclassified samples, $\mathbf{z}_i$ represents the logits for sample i, and Top-$k(\mathbf{z}_i)$ denotes the Top-k logits of sample i.

When a sample's variance of logits is smaller than σ^2_{mis}, it is considered a hard sample.

Our mechanism is proposed with the consideration of marginal effects, ensuring that not all samples are identified as hard samples. As the number of hard samples increases, the marginal accuracy gain diminishes, while time and resource consumption increase. To avoid unnecessary computation, we employ an auto detection mechanism to selectively identify genuine hard samples, thereby optimizing resource usage and improving accuracy.

3.3 Rerank for Hard Sample

Inspired by Vulić et al. (2021), we conceptualize the classification task as a sentence similarity pairing task. This perspective allows us to consider semantic labels as short texts. According to the perspective of multi-view learning, the same entity can be described from multiple approaches or perspectives, with each different description constituting a distinct view of the entity (Li et al., 2019). The semantic distance between text and semantic labels can be viewed as an alternative perspective on the logits of classification. This rationale provides sufficient grounds for reranking logits based on semantic distance. Given a hard sample *S* and all semantic labels {l_1,l_2,...,l_n}, we calculate their semantic embedding e_s and $e_l = \{e_{l_1}, e_{l_2}, ..., e_{l_n}\}$ by fine-tuned Text Embedding. Then, their Cosine Similarity will be used as the weight of logits for reranking. The calculations of the reranking are as follows:

$$\boldsymbol{v_s} = \text{MLP}(\text{PLM}(S)), \tag{8}$$

$$\boldsymbol{D_s} = \text{Sim}(\boldsymbol{e_s}, \boldsymbol{e_l}), \tag{9}$$

$$\boldsymbol{R_s} = \boldsymbol{D_s} \cdot \text{softmax}(\boldsymbol{v_s}), \tag{10}$$

where v_s denotes the classification logits computed by MLP-based PLM, D_s denotes the semantic distance between the hard sample and semantic labels, and R_s denotes the logits of the hard sample after reranking.

4 Experimental Setup

4.1 Datasets

CMCC-34[1]: This is a long-text intent detection dataset for Chinese multi-turn customer service dialogues. It comprises 34 classes, totaling 20,000 samples, with an average length of 379 tokens per sample. This dataset contains significant amounts of noise due to being transcribed from speech. It is considered a relatively realistic dataset in the field of intent recognition. Detailed studies on this dataset have been conducted by Xu et al. (2022) and Huang et al. (2024). More details are shown in Appendix A.

iFlyTek-119[2]: The dataset contains more than 17,000 Chinese long text annotation data about app application descriptions, including various application topics related to daily life, with a total of 119 categories. It is available as part of the recently published CLUE benchmark (Xu et al., 2020).

Banking-77[3]: The dataset provides a very fine-grained set of intents in a banking domain. It comprises 13,083 customer service queries labeled with 77 intents. It focuses on fine-grained single-domain intent detection. It is available as part of the recently published DialoGLUE benchmark (Mehri et al., 2020).

4.2 Hyperparameters and Baselines

We train each model with five different random seeds, and the accuracy is averaged. The batch size is set to 24, and early stopping is employed with a patience strategy of 4. To prevent overfitting, dropout with a probability of 0.1 is applied. The parameters are updated using the Adam algorithm, with the learning rate initialized to 2e-5.

As a model-agnostic post-processing method, we apply it across seven models. **BERT-base**: A highly popular pre-trained language model (Devlin et al., 2019). **RoBERTa-base**: An improved variant of BERT, pre-trained with more data (Liu et al., 2019). We use RoBERTa as the backbone of TextEmbedding. **CONVBERT**: Fine-tune BERT on a large open-domain dialogue corpus with 700 million conversations (Mehri et al., 2020). **APHAN**: An adjacency pairsaware hierarchical attention Network for dialogue intent classification (Xu et al., 2022). **HLDIC**: An Hierarchical Label-Aware Dialog Intent Classification network (Huang et al., 2024). We also apply our methods on LLMs, **Llama2-7B** (Touvron et al., 2023) and **Mistral-7B** (Jiang et al., 2023). Both LLMs integrate Quantized Low-Rank Adapters (QLoRA), a supervised fine-tuning method that significantly reduces memory usage during training (Dettmers et al., 2023).

[1] http://www.cips-cl.org/static/CCL2018/call-evaluation.html

[2] https://github.com/CLUEbenchmark/CLUE

[3] https://huggingface.co/datasets/banking77

Model	CMCC-34		iFlyTek-119		Banking-77	
	Origin	**Rerank**	Origin	**Rerank**	Origin	**Rerank**
BERT-base (Devlin et al., 2019)	56.53	**59.44**	60.29	**62.58**	92.61	**93.66**
RoBERTa-base (Liu et al., 2019)	57.92	**59.90**	60.31	**62.65**	93.45	**94.02**
CONVBERT (Mehri et al., 2020)	-	**-**	-	**-**	92.67	**93.79**
AP-HAN (Xu et al., 2022)	57.77	**60.49**	60.31	**62.46**	-	**-**
HLDIC (Huang et al., 2024)	58.36	**60.59**	-	**-**	-	**-**
Llama2-7B(QLoRA) (Touvron et al., 2023)	57.63	**60.22**	60.42	**62.04**	93.47	**94.10**
Mistral-7B(QLoRA) (Jiang et al., 2023)	59.30	**61.65**	59.31	**61.42**	93.05	**94.00**

Table 1: Experiment results on the full test sets. **Origin** denotes the accuracy without using our method, **Rerank** represents the accuracy after using our LRSL. '**-**' indicates that the model is not suitable for the datasets, such as CONVBERT for the two Chinese datasets CMCC-34 and iFlyTek-119, AP-HAN for single-sentence datasets like Banking-77, and HLDIC for datasets other than CMCC-34 due to its hierarchical labels.

Model	CMCC-34			iFlyTek-119			Banking-77		
	Samples	Origin	**Rerank**	Samples	Origin	**Rerank**	Samples	Origin	**Rerank**
BERT-base (Devlin et al., 2019)	44.53%	37.42	**43.95**	40.12%	38.73	**39.71**	9.76%	56.00	**66.74**
RoBERTa-base (Liu et al., 2019)	44.05%	38.94	**43.36**	39.62%	38.92	**40.20**	9.15%	58.54	**64.77**
CONVBERT (Mehri et al., 2020)	-	-	**-**	-	-	**-**	8.56%	54.51	**67.40**
AP-HAN (Xu et al., 2022)	44.33%	38.60	**44.74**	40.77%	38.50	**41.07**	-	-	**-**
HLDIC (Huang et al., 2024)	35.03%	36.29	**42.66**	-	-	**-**	-	-	**-**
Llama2-7B(QLoRA) (Touvron et al., 2023)	47.88%	40.05	**45.54**	46.23%	44.09	**47.50**	15.92%	71.49	**75.36**
Mistral-7B(QLoRA) (Jiang et al., 2023)	46.95%	40.02	**44.97**	46.19%	42.21	**44.05**	17.22%	71.48	**76.92**

Table 2: Experiment results on the hard samples test sets. **Samples** represents the proportion of the hard samples in the test set. Due to the test set answers of the iFlyTek-119 dataset being closed, we use the validation set for our hard sample experiments.

We utilize quantization with 4-bit precision, enabling training on a single Nvidia RTX3090. More implementation details are shown in Appendix B.

5 Results and Discussion

5.1 Main Experiments

5.1.1 Experiment on Full Samples

The main experimental results on full samples shown in Table 1 demonstrate that our post-processing method achieved varying degrees of improvement across different models and datasets, with particularly significant performance observed on the CMCC dataset, surpassing the original state-of-the-art (SOTA) result (58.36%) by up to 3.29%. Compared with the other two datasets, CMCC has the longest average length and the largest amount of noise. This shows to a certain extent that our method is more effective in processing confusing samples. BERT was first proposed among all displayed models with the lowest accuracy, but after using LRSL, it can surpass the current SOTA.

On the iFlyTek-119 dataset, LLama and Mistral do not outperform other PLMs. This is likely due to the nature of the text in the dataset, which consists of app comments rather than dialogue-focused content. The training data used in the supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) stages for these LLMs predominantly comprises question-answering datasets. Consequently, these models are optimized for dialogue recognition and intention classification tasks.

Although the Banking-77 dataset has a relatively short average token length and consists of single sentences, the large number of intents still leads the model to identify hard samples during classification. Our method is effective not only for hard samples in long texts but also for those in short texts. This demonstrates that our approach is relatively universal and not limited to long texts.

5.1.2 Experiment on Hard Samples

Further analysis of hard samples shown in Table 2 reveals that our method exhibits more prominent performance under such conditions. Due to the automatic detection, each model has its own hard samples. Our method shows more significant improvements on different datasets, with the highest improvement reaching 6.53% on the CMCC dataset, 3.41% on the iFlyTek dataset, and 12.89% on the Banking77 dataset. This proves that

Model	CMCC-34	iFlyTek-119	Banking-77
BERT-base-***LRSL***	59.44	62.58	93.66
✘ triplets	59.09	61.65	93.54
✘ fine-tune	56.46	61.65	92.59
RoBERTa-base-***LRSL***	59.90	62.65	94.02
✘ triplets	59.74	62.31	93.94
✘ fine-tune	57.85	60.96	93.43
AP-HAN-***LRSL***	60.49	62.46	-
✘ triplets	60.33	62.23	-
✘ fine-tune	57.67	60.00	-
HLDIC-***LRSL***	60.59	-	-
✘ triplets	60.40	-	-
✘ fine-tune	58.36	-	-
CONVBERT-***LRSL***	-	-	93.79
✘ triplets	-	-	93.29
✘ fine-tune	-	-	92.67
Llama2-7B-***LRSL*** (QLoRA)	60.22	62.04	94.10
✘ triplets	59.45	61.88	93.93
✘ fine-tune	57.65	60.35	93.54
Mistral-7B-***LRSL*** (QLoRA)	61.65	61.42	94.00
✘ triplets	60.98	61.15	93.90
✘ fine-tune	59.43	59.54	93.05

Table 3: Ablation experiment of Semantic Mapping, where ✘ **triplets** represents fine-tuning Text Embedding without triplets and ✘ **fine-tune** represents utilizing Text Embedding without fine-tuning.

focusing on the hard samples to rerank aligns with the principle of marginal utility, saving resources and concentrating on improving the accuracy of hard samples.

5.1.3 LRSL Fits LLMs Well

In the scenario of limited GPU resources, we conduct fine-tuning experiments of QLoRA with 4-bit quantization on the LLMs, which may lead to accuracy degradation. However, by applying our method, we successfully compensate for this drawback and achieve substantial improvements. Our experiments shown in Table 1 and 2 prove the universality and effectiveness of our proposed approach. A small-parameter Text Embedding can guide LLMs to make better choices.

5.2 Ablation Study

In this section, we conduct ablation experiments to investigate the effects of **Semantic Mapping** and **Hard Sample Auto Detection**.

5.2.1 Effects on Semantic Mapping

The ablation experiment of Semantic Mapping results in Table 3 show that the Text Embedding fine-tuning and the triplets in our method significantly improve semantic relevance, providing substantial guidance when dealing with hard samples.

It can be observed that removing the fine-tuning for Text Embedding results in a decline in accuracy

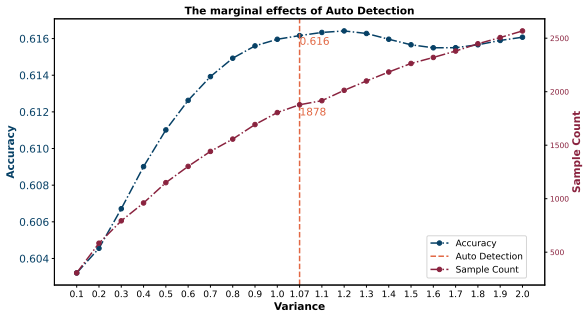


Figure 4: Marginal effects of Auto Detection on the CMCC-34 with Mistral. The rest is shown in Appendix D. The blue line denotes accuracy, the red line denotes the number of hard samples (understood as the consumption of time), and the orange line denotes the choice of Auto Detection.

across seven different models on three datasets. Taking BERT on CMCC-34 as an example, the model decreases by approximately 2.98% in accuracy. The significant decrease demonstrates that fine-tuning injects substantial prior knowledge, allowing the text embedding to construct a comprehensive semantic space.

We fine-tune Text Embedding without triplets, which results in a decrease in accuracy across the three datasets. Fine-tuning with triplets primarily helps to distinguish between confused and true labels, refining the semantic space for more accurate representation. A comprehensive and accurate semantic space is the reason for the success of LRSL. Furthermore, we visualize the effects of Text Embedding in Appendix C.

5.2.2 Effects on Hard Sample Auto Detection

To explore whether the Auto Detection mechanism can optimize resource consumption while maximizing accuracy, we conducted a marginal effects experiment. This experiment aims to evaluate the effectiveness of the Auto Detection in balancing the trade-off between resource usage and accuracy.

As depicted in Figure 4, when the number of hard samples increases (green line), the time spent also increases. Initially, the accuracy (blue line) improves, indicating a benefit from the inclusion of additional hard samples. However, this improvement diminishes over time, suggesting diminishing returns on additional resource investment. The orange line demonstrates that the Auto Detection mechanism selects an appropriate variance as the criterion for identifying hard samples, thereby approaching the point of optimal marginal effect.

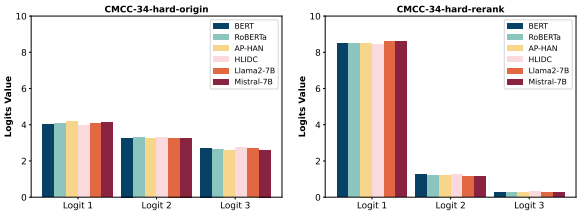


Figure 5: Hard Samples of Top-3 classification logits on CMCC-34. The figures compare the logits of multiple models before and after reranking. The logits values are scaled to a range from 0 to 10.

Model	CMCC-34	iFlyTek-119	Banking-77
Similarity-based	57.45	58.25	93.39
MLP-based	**57.92**	**60.31**	**93.45**

Table 4: Similarity-based vs. MLP-based on full samples. Their base model is RoBERTa-base.

Our mechanism enables the model to identify suitable hard samples for reranking, rather than applying reranking to all samples. This ensures that accuracy and time consumption are optimized near the point of maximal marginal effect.

5.3 Further Discussion

Visualization of Logits Reranking. To explore the impact of Rerank on the classification logits, we visualize the contrast between the variances of logits before and after reranking on CMCC-34, which is shown in Figure 5. The visualizations for the remaining two datasets are included in Appendix E. After reranking, the confidence in selecting a particular class for the logits is significantly enhanced, as evidenced by the increase in variance values.

Similarity-based vs. MLP-based. There may be a question "Why not simply use similarity for classification?", the experiment results in Table 4 prove that similarity-based classification does not perform as well as MLP-based classification on full datasets, especially for easy samples shown in Table 5. This is why we only use the semantic information of Text Embedding as an enhancement step for difficult samples. Fine-tuned Text Embedding has better discriminative ability and can guide MLP-based to perform better classification.

Labels with Less Semantic. To validate the effectiveness of LRSL in scenarios where label semantics are limited or the number of labels is small, we conduct an additional experiment using the Tweet Sentiment Extraction Classification (TSEC) Dataset[4], which consists of only three labels: 0-Negative, 1-Neutral, and 2-Positive.

Model	CMCC-34	iFlyTek-119	Banking-77
Similarity-based	71.32	73.30	96.91
MLP-based	**72.79**	**75.76**	**97.09**

Table 5: Similarity-based vs. MLP-based on easy samples. Their base model is RoBERTa-base.

Model	Origin	**Rerank**
BERT-base (Devlin et al., 2019)	79.43	**80.09**
RoBERTa-base (Liu et al., 2019)	79.91	**80.32**
CONVBERT (Mehri et al., 2020)	80.50	**81.21**
Llama2-7B(QLoRA) (Touvron et al., 2023)	78.21	**79.82**
Mistral-7B(QLoRA) (Jiang et al., 2023)	78.98	**80.16**

Table 6: TSEC experiment on the full test sets. **Origin** denotes the accuracy without using our method, and **Rerank** represents the accuracy after using our LRSL.

Considering the limited semantic richness of the labels, we can employ a large language model to augment the label semantics by transforming the original labels into more distinctive and descriptive terms. This semantic enhancement allows for more nuanced differentiation among the labels, potentially improving model performance on tasks with limited label diversity.

As shown in Tables 6 and 7, LRSL further improves the classification performance of all models, with a maximum improvement of 1.61%. Additionally, LRSL enables the models to adaptively select hard samples, resulting in significant performance gains even for these challenging cases. This demonstrates that our method remains effective even when label semantics are lacking.

5.4 Case Study

In this section, we select a hard sample of CMCC-34 detected by BERT for a case study, which is shown in Figure 6. Since the samples in CMCC-34 are Chinese speech-to-text transcriptions, we have translated them into English, corrected any typos, and added roles such as customer service and user for better presentation. These annotations are not present in the original dataset.

In Figure 6, we can observe that the original Top-3 classification logits for this hard sample are as follows: "Inquiry Queries Account Information (IQAI)", "Complaint Grievances Billing Issues (CGBI)", and "Complaint Grievances Service Processing Issues(CGSPI)". Their respective logits are

[4] https://huggingface.co/datasets/mteb/tweet_sentiment_extraction

Hard Sample

S: 有没有您好很高兴为您服务请讲 **U:** 你说你不用管喂你好我想问一下刚才我给别人充了三十的话费人家怎么收到二十九块九毛四呀 **S:** 您稍等我帮您看一下 **U:** 好的吧 S: 他是这样的人家到账的话是属于三十块钱他是就是说您这个充值话费是交了三十是享受六块钱的总共但是我们给您充了是三十块钱总它是一个二十九块九毛四还有一个六分我把这个充值记录发给您您可以看一下好吧 **U:** 好的

[Translation] **S:** Hello, I'm happy to help you. Please go ahead. **U:** You said... never mind. Hello, I'd like to ask why, when I recharged someone else's phone with 30 yuan, they only received 29.94 yuan? **S:** Please hold on, I'll check that for you. **U:** Okay. **S:** It's like this: the person received 30 yuan, but when you recharge with 30 yuan, there is a fee of 6 cents, so the total credited amount is 29.94 yuan. I will send you the recharge record, and you can check it. Is that okay? **U:** Okay.

Tok-3 Logits ↓	Semantic Distance	Reranked Logits ↓
IQAI: **4.700** ✗	IQAI: **0.023**	CGBI: **0.042** ✓
CBGI: **4.677**	CGBI: **0.328**	CGSPI: **0.006**
CGSPI: **2.705**	CGSPI: **0.342**	IQAI: **0.0002**

Figure 6: Case study. Due to space constraints, only the Top-3 logits are displayed. **S** represents the Customer Service, while **U** represents the User. Incorrect intents are marked in red, and correct intents are marked in green.

Model	Samples	Origin	**Rerank**
BERT-base (Devlin et al., 2019)	25.07%	55.81	**58.10**
RoBERTa-base (Liu et al., 2019)	29.00%	61.34	**62.74**
CONVBERT (Mehri et al., 2020)	25.08%	57.96	**60.75**
Llama2-7B(QLoRA) (Touvron et al., 2023)	30.39%	59.59	**64.90**
Mistral-7B(QLoRA) (Jiang et al., 2023)	31.52%	61.58	**65.35**

Table 7: TSEC experiment on the hard samples test sets. **Samples** represents the proportion of the hard samples in the test set.

very close, especially the first two, which are 4.700 and 4.677. The small variance indicates that the model is not confident in distinguishing the correct intent.

We then calculate the semantic distance between the hard sample and these labels, which will serve as weights for the logits. We can see that the largest original logit has a semantic distance of 0.023, indicating that, from a semantic perspective, the sample lacks a clear relationship with the intent "IQAI". After reranking based on these semantic distances, the largest logit is reordered to a lower position, while the second-ranked correct intent is promoted to the first position. It can be observed that LRSL effectively leverages semantic labels to increase the model's classification confidence, thereby improving the accuracy.

6 Conclusion

We propose LRSL, a model-agnostic post-processing method that leverages label semantics and auto detection of hard samples to improve classification accuracy. Our method demonstrates that label semantics serve as an effective reranker for hard samples in MLP-based classification. By reranking classification logits, LRSL provides a plug-and-play solution that does not interfere with the original model's training process. The results demonstrate the effectiveness of our method.

7 Acknowledgements

This work was supported by the National Natural Science Foundation of China (71472068 and 62306119) and the Natural Science Foundation of Guangdong Province (2021A1515011864).

Limitations

To automatically detect the hard samples, we choose the distribution's variance of Top-k classification logits as the metric. However, this approach requires a separate dataset from the training set or an examination of the validation set's distribution, and it is not sufficiently comprehensive to fully measure the difficulty of samples. Future work will focus on adopting additional metrics to automatically identify hard samples.

To construct an accurate semantic space for LRSL, we chose to use an additional Text Embedding model for fine-tuning. While this introduces extra parameters, it was selected for its plug-and-play capability. In the future, this embedding model can be replaced with more advanced Text Embedding models as they become available.

References

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 830–835. AAAI Press.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

John Fields, Kevin Chovanec, and Praveen Madiraju. 2024. A Survey of Text Classification With Transformers: How Wide? How Large? How Long? How Accurate? How Expensive? How Safe? *IEEE Access*, 12:6518–6531.

Radhika Gaonkar, Heeyoung Kwon, Mohaddeseh Bastan, Niranjan Balasubramanian, and Nathanael Chambers. 2020. Modeling label semantics for predicting emotional reactions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4687–4692, Online. Association for Computational Linguistics.

Simin Huang, Peijie Huang, Yuhong Xu, Jingzhou Liang, and Jingde Niu. 2024. Exploring label hierarchy in dialogue intent classification. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11511–11515.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Yingming Li, Ming Yang, and Zhongfei Zhang. 2019. A survey of multi-view representation learning. *IEEE Trans. Knowl. Data Eng.*, 31(10):1863–1883.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601, Singapore. Association for Computational Linguistics.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tür. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *CoRR*, abs/2009.13570.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Aaron Mueller, Jason Krone, Salvatore Romeo, Saab Mansour, Elman Mansimov, Yi Zhang, and Dan Roth. 2022. Label semantic aware pre-training for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8318–8334, Dublin, Ireland. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Yangqiu Song and Dan Roth. 2014. On dataless hierarchical text classification. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1579–1585. AAAI Press.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ivan Vulić, Pei-Hao Su, Samuel Coope, Daniela Gerz, Paweł Budzianowski, Iñigo Casanueva, Nikola Mrkšić, and Tsung-Hsien Wen. 2021. ConvFiT: Conversational fine-tuning of pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1151–1168, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

Jiabao Xu, Peijie Huang, Youming Peng, Jiande Ding, Boxi Huang, and Simin Huang. 2022. Adjacency Pairs-Aware Hierarchical Attention Networks for Dialogue Intent Classification. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7622–7626, Singapore, Singapore. IEEE.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

A Details of Datasets

In this section, we provide detailed information about the datasets used in our experiments. The datasets include CMCC-34, iFlyTek-119, and Banking-77. The following Table 8 summarizes the split of the train, development (dev), and test sets, along with the average token length and the number of intents for each dataset.

The CMCC-34 and iFlyTek-119 datasets have significantly longer average token lengths compared to the Banking-77 dataset. Specifically, the average token length for CMCC-34 is 379, and for iFlyTek-119 is 276, whereas the average token length for Banking-77 is only 14. This indicates that CMCC-34 and iFlyTek-119 contain more complex and verbose text data.

As shown in Figure 7, the label distribution in the training sets of CMCC-34 and iFlyTek-119 is imbalanced, exhibiting a long-tail distribution, which makes these datasets more challenging for classification tasks. In contrast, the Banking-77 dataset has a relatively balanced label distribution, simplifying the classification task compared to the other two datasets.

B MLP-based Implementation Details

B.1 Non-LLM Implementation Details

We conducted experiments with five non-LLMs: BERT, RoBERTa, CONVBERT, AP-HAN, and

Dataset	Split of train/dev/test	Average Token Length	Label
CMCC-34	12799 / 3200 / 4000	379	34
iFlyTek-119	12133 / 2599 / 2600	276	119
Banking-77	8011 / 2006 / 3084	14	77

Table 8: Details of CMCC-34, iFlytek-119, and Banking-77 datasets.

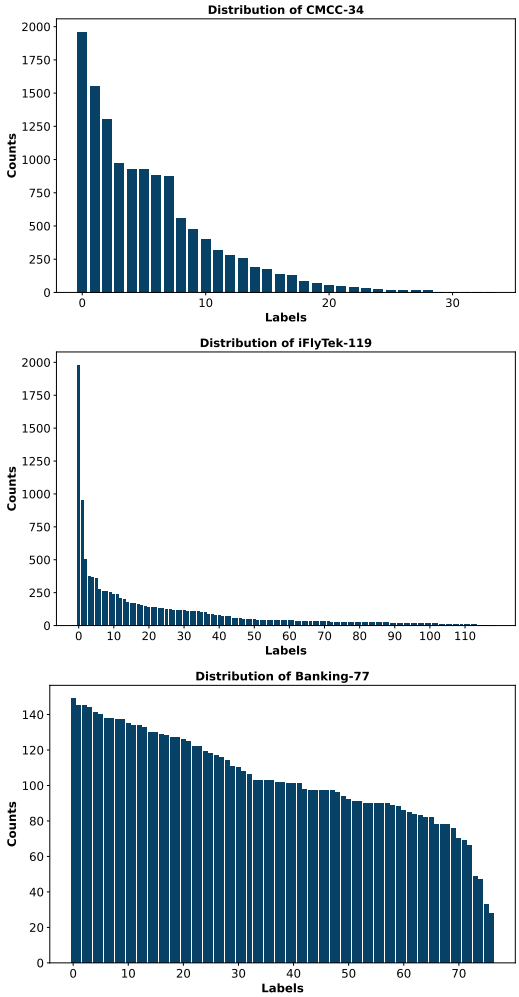


Figure 7: Label distribution of 3 datasets.

HLDIC, all of which are autoencoding models. For the non-LLMs, we adopt the relatively simple model structure illustrated in Figure 8 to focus on exploring the effectiveness of our method. The model first encodes the text using PLMs, extracting the last hidden state of the $[CLS]$ token. These representations are then passed through a Dropout layer to reduce the risk of overfitting. Finally, they are fed into an MLP layer for classification.

B.2 LLM Implementation Details

For our experiments, we utilize two LLMs: Mistral-7B-Instruct-v0.2 and Llama-2-7B-Chat. The training hyperparameters for both models were carefully selected to ensure optimal performance. We set the learning rate to 1e-5, with a weight decay of 0.001, and a warmup ratio of 1.0. The maximum gradient norm was capped at 1.0 to prevent gradient explosion, and a dropout rate of 0.1 was used to mitigate overfitting.

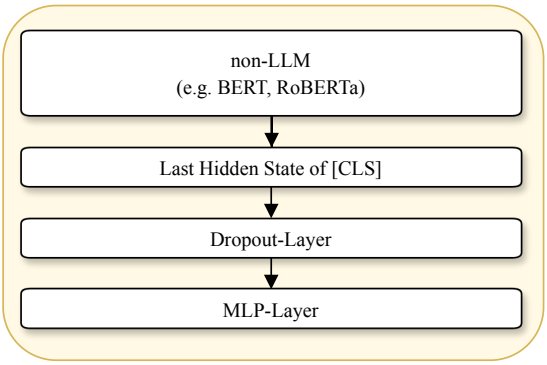


Figure 8: Implementation Details of non-LLM.

For LoRA, we configure the LoRA rank (r) to 8 and the LoRA alpha to 32. The models are trained with a batch size of 8 over 5 epochs. These hyperparameters are chosen to balance training efficiency and model performance, ensuring that the models can learn effectively without overfitting. The warmup ratio of 1.0 implies that the learning rate linearly increases from 0 to its maximum value during the entire first epoch, which helps stabilize the training process. The LoRA parameters are set to allow efficient fine-tuning by injecting task-specific adaptation layers with minimal additional parameters.

C Semantic Distance Between Labels

We visualize the effects of Text Embedding and compare the results before and after fine-tuning. The effects of fine-tuning is shown in Figure 9. The heat maps are quite significant: semantic distance between labels has a clear distinction boundary, which means Text Embedding already has the ability to distinguish confusing samples. This demonstrates that Text Embedding enables clear separation between distances among labels, providing sufficient guidance for the model to rerank hard samples effectively.

D Marginal Effects on Auto Detection

As shown in Figure 10, we conduct marginal effects experiments on CMCC-34 and Banking-77. Because the answers of the test set of iFlyTek-119 dataset are retained by the benchmark platform, we can not access them and thus preclude

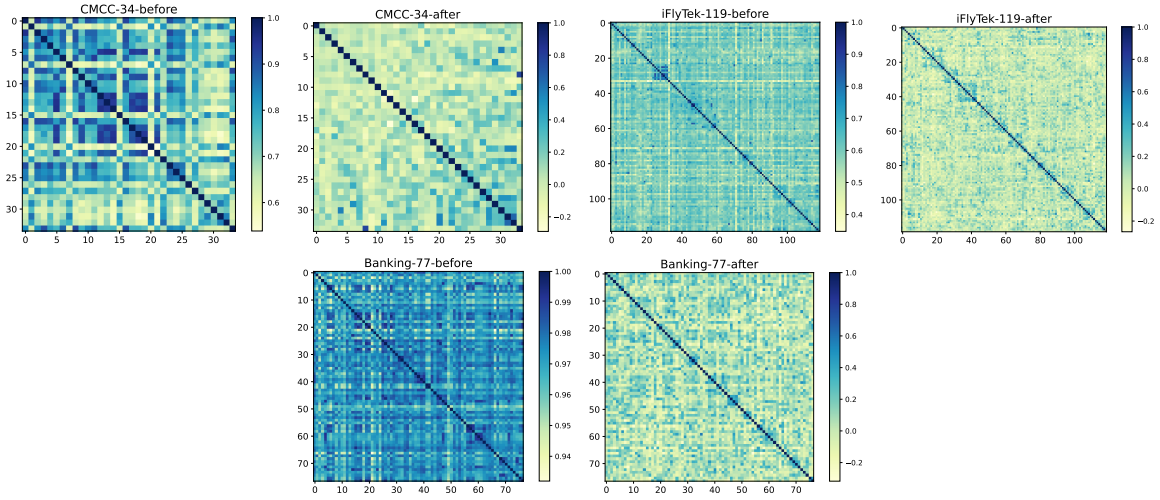


Figure 9: Semantic distances between labels on CMCC-34, iFlyTek-119, and Banking-77 datasets. The figures compare the semantic distances of Text Embedding before and after fine-tuning. A larger value indicates a closer distance.

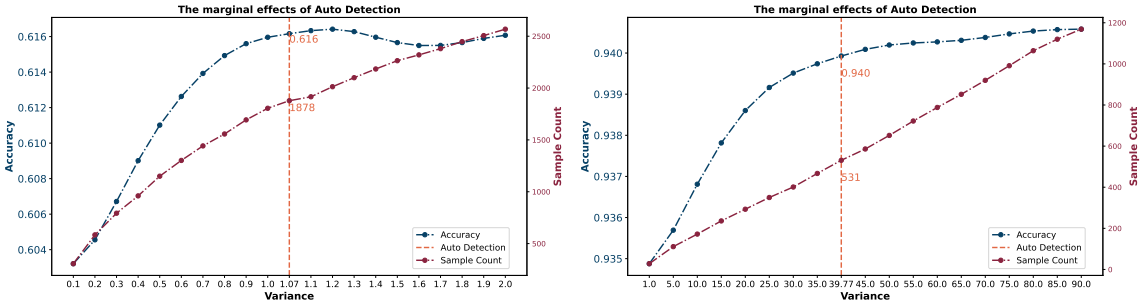


Figure 10: Marginal effects of Auto Detection on the CMCC-34 and Banking-77 with Mistral. The blue line denotes accuracy, the red line denotes the number of hard samples (understood as the consumption of time), and the orange line denotes the choice of Auto Detection.

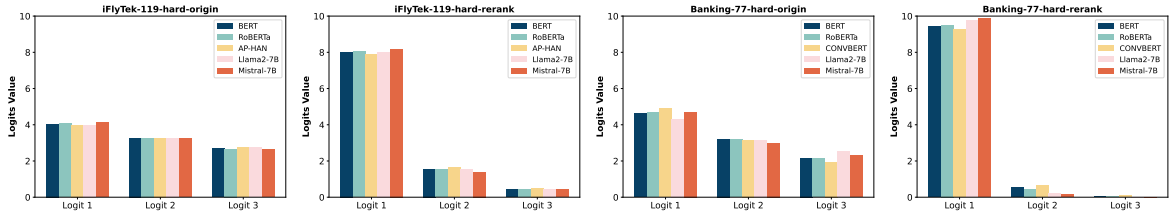


Figure 11: Hard samples of Top-3 classification logits on iFlyTek-119 and Banking-77 datasets. The figures compare the logits of multiple models before and after reranking. The value of logit is scaled to a range from 0 to 10.

experimentation on this dataset. Our Auto Detection mechanism can adaptively select variance for choosing hard samples on both datasets, eliminating the need for manually setting thresholds for each dataset. This capability has the potential to generalize across diverse datasets.

E Visualization of Logits Reranking

In this section, we further visualize the effects of Logits Reranking on another two datasets: iFlyTek-119 and Banking-77, which can be shown in Figure 11. After reranking, the classification logits of hard samples shift from a relatively even distribution to a pronounced preference for the first probability, indicating a significant increase in the model's classification confidence. This adjustment leads to a stronger inclination towards confidently selecting a single label rather than being indecisive.