Rationale-Guided Distillation for E-Commerce Relevance Classification: Bridging Large Language Models and Lightweight Cross-Encoders

Sanjay Agrawal∗
Amazon, India
sanjagr@amazon.com

Faizan Ahemad∗
Amazon, India
ahemf@amazon.com

Vivek Sembium
Amazon, India
viveksem@amazon.com

Abstract

Accurately classifying the relevance of Query-Product pairs is critical in online retail stores such as Amazon, as displaying irrelevant products can harm user experience and reduce engagement. While Large Language Models (LLMs) excel at this task due to their broad knowledge and strong reasoning abilities. However, their high computational demands constrain their practical deployment in real-world applications. In this paper, we propose a novel distillation approach for e-commerce relevance classification that uses "rationales" generated by LLMs to guide smaller cross-encoder models. These rationales capture key decision-making insights from LLMs, enhancing training efficiency and enabling the distillation to smaller cross-encoder models deployable in production without requiring the LLM. Our method achieves average ROC-AUC improvements of 1.4% on 9 multilingual e-commerce datasets, 2.4% on 3 ESCI datasets, and 6% on GLUE datasets over vanilla cross-encoders. Our 110M parameter BERT model matches 7B parameter LLMs in performance (< 1% ROC-AUC difference) while being 50 times faster per sample.

1 Introduction

Large-scale e-commerce search systems, used by companies like Amazon and Walmart, typically follow a multi-step process to retrieve relevant products for a given query (Guo et al., 2022). The process starts with an initial retrieval step that generates a broad match set for the query. A relevance model is then applied to capture the nuanced relationship between the customer's query intent and the products in this match set (Momma et al., 2022). This relevance model plays a role similar to reranker models used in Retrieval-Augmented Generation (RAG) pipelines. In real-time retrieval tasks, user queries are matched to products as they occur. However, latency and computational constraints often limit the complexity of matching algorithms, resulting in reduced accuracy and coverage. To mitigate this, search engines pre-generate product sets offline for frequently searched queries, storing them in production tables. This offline retrieval process, which powers the majority of search operations, combines lexical, behavioral, and semantic retrieval models to return a wide range of results. Once the offline retrieval is complete, the focus shifts to refining this broad set of results to better align with the customer's intent. This is achieved using a **relevance model**, where lightweight cross-encoder models (Mangrulkar et al., 2022) are typically employed to filter out poor <query, product> pairs, ensuring a high-quality user experience. In this paper, we focus on building high-performing cross-encoder relevance models to predict the relevance of <query, product> pairs. Given that this relevance model needs to evaluate millions of pairs daily, it must be small and efficient language model to minimize compute costs and inference time while maintaining high accuracy.

The advent of LLMs has revolutionized relevance classification and retrieval tasks. LLMs excel in these tasks due to their extensive pretraining, which equips them with vast knowledge, enabling high precision in classification. A key breakthrough in this area is the introduction of "rationales" or "chains of thought"—representing the cognitive processes or contextual understanding that the model uses to arrive at specific decisions or classifications. However, the impressive capabilities of LLMs come at the cost of immense computational demands, far exceeding those of cross-encoder models, making them impractical for large-scale prediction tasks. For instance, classifying 50 million <Query, Product> pairs using a 20B parameter LLM can take several days on a single GPU, while cross-encoder models can accomplish

∗ These authors contributed equally to this work.

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 136–148
January 19–24, 2025. ©2025 Association for Computational Linguistics

the same task in just a few hours, underscoring the computational efficiency of cross-encoders.

Drawing inspiration from industry practices and their inherent challenges, we introduce a novel approach to enhance relevance classification by harnessing the reasoning capabilities of LLMs to boost the performance of cost-effective cross-encoder models. Our method integrates LLM-generated rationales as an auxiliary task during cross-encoder training, utilizing a cross-encoder-decoder architecture. In this framework, the cross-encoder handles the primary binary classification task, while the decoder generates rationales based on LLM outputs. For inference, we streamline the process by deploying only the cross-encoder model, removing the need for the LLM-based rationale generator or decoder module. This design ensures an efficient and powerful relevance classification system that balances performance with computational efficiency. Below we summarize our **key contributions:**

1. Develop a novel cross-encoder-decoder architecture (Sec. 4.3) that leverages LLM-generated reasoning to enhance relevance classification. This approach combines the reasoning capabilities of LLMs with the efficiency of cross-encoders, utilizing LLM written rationales as auxiliary training data while maintaining low inference costs by deploying only the cross-encoder at runtime.

2. Rigorous evaluation on diverse datasets, including 9 multilingual e-commerce datasets, 3 ESCI datasets, and public datasets (GLUE and QADSM). Our method achieves average ROC-AUC improvements of 1.4% on 9 multilingual e-commerce datasets, 2.4% on 3 ESCI datasets, 6% on GLUE datasets over vanilla cross-encoders and state-of-the-art performance on QADSM surpassing finetuned LLMs. Our 110M parameter model matches 7B parameter LLMs in performance ($<$ 1% ROC-AUC difference) while being $50x$ faster per sample.

3. We conduct several ablations (Sec. 5.2) to examine how rationale distillation benefits the model across different data sizes. Our rationale-guided distillation, utilizing LLM-generated reasons, helps the model focus on relevant tokens and improves attention between query and product title tokens. With 10K samples, our method outperforms the best fine-tuned LLM in 6/9 cases, and with 100K samples, it remains competitive, outperforming in 1/9 cases.

2 Related Work

Knowledge distillation (Agrawal et al., 2023b) (Hinton, 2015) (KD) focuses on training a smaller and inexpensive student model to replicate the behavior of a larger, complex teacher model by minimizing a distillation loss based on the teacher's soft target probabilities. The key advantage is that soft probabilities contain richer information than hard labels. KD for Transformer models has been widely studied (Freitag et al., 2017). Since the introduction of BERT (Devlin et al., 2018), efforts to distill these models have led to variants like DistilBERT (Sanh, 2019), TwinBERT (Lu et al., 2020), MobileBERT (Sun et al., 2020), and MiniLM (Wang et al., 2020). Among these, DistilBERT, with its 6-layers, is the most commonly used for its balance of performance and efficiency.

Large language models (LLMs) have significantly larger parameter spaces than pre-trained models (Zhao et al., 2023) like BERT (Devlin et al., 2018), often in the billions. For instance, GPT-3 (Brown et al., 2020) has around 175B parameters, while Megatron-Turing NLG (Smith et al., 2022) boasts 530B. Deploying these large models is challenging due to their high computational and memory demands, prompting practitioners to use smaller, distilled models (Gu et al., 2023). For example, Hsieh et al. (Hsieh et al., 2023) demonstrate the effectiveness of distilling LLM-generated rationales using T5 (Raffel et al., 2020) (Agrawal et al., 2023a) encoder-decoder models in a text-to-text framework. In contrast, our work applies this concept to smaller BERT cross-encoder models (110M parameters) in a classification setting, outperforming few-shot prompted LLMs and even surpassing fine-tuned LLMs in certain e-commerce relevance classification tasks.

3 Problem Statement

We develop a high-performing cross-encoder relevance model R to predict the relevance of a <query, product> pair. The model is trained on human-annotated query-product pairs, $D^{QP}_{label} = \{(q_i, p_i, y_i)\}^{n}_{i=1}$, where q_i, p_i, and y_i represent the query, product title, and a binary relevance label. We also define an LLM θ_{LLM} that, using a designed prompt $\tau(q_i, p_i, y_i)$, determines the reasoning behind relevance. The goal is to leverage LLM reasoning to improve the efficiency of model R.

4 Proposed Method

Our approach aims to distill the reasoning capabilities of LLMs into our cost-efficient cross-encoder model to improve its performance. We first investigate the capabilities of LLMs for our task and then devise a method to distill this capability. The key steps are outlined below:

1. **Relevance Classification via Direct Answering from LLM**: We first use an LLM to determine the relevance of a query-product pair in natural language, tested in both zero-shot (ZS) and few-shot (FS) settings.

2. **LLM Finetuning with Linear Layer**: We finetune a linear layer on top of the LLM's last layer output using our training data. This serves as our "Maximum achievable performance" or "Performance ceiling" which is the highest performance possible to achieve using Language modelling. Our production model (fine-tuned BERT) is only 110M parameter model, compared to the 7B LLM used here, is over 50x faster per sample.

3. **Reasoning Generation and Distillation into Cross-Encoder**: We generate reasoning for relevance using the LLM and use reasoning generation as an auxiliary task for training the cross-encoder model. This is achieved through a cross-encoder-decoder architecture where the BERT cross-encoder handles the primary binary classification task, and the decoder generates reasoning mimicking the LLM's rationale.

4.1 Relevance Classification via Direct Answering from LLM

We use an LLM to assess the relevance of a query-product pair in both zero-shot and few-shot settings, providing a baseline for the LLM's capabilities.

In the zero-shot setting, the LLM is tasked with determining the relevance without any prior specific training on similar query-product relevance tasks. The input to the LLM is formatted using template as: "[Query] is [Product Title] relevant?" The output is parsed to classify as 'relevant' if the response starts with 'Yes', 'not relevant' if response starts with 'No'. **Example:**

```
Query: wireless mouse
Product Title: Logitech MX Master 3
Advanced Wireless Mouse
LLM Response: Yes, this product is
relevant to the query.
```

In the few-shot setting, the LLM is provided with a few annotated examples before making a relevance determination. This approach leverages the model's in-context learning ability. Each example is presented in the same format as the zero-shot queries but includes examples at the beginning.

Example:

```
In-Context Examples:
1. Query: "wireless mouse"
   Product Title: "Logitech MX Master 3
   Advanced Wireless Mouse"
   Relevance: "Yes"
2. Query: "gaming keyboard"
   Product Title: "Corsair K95 RGB Platinum
   Mechanical Gaming Keyboard"
   Relevance: "Yes"
Target:
Query: "wireless mouse"
Product Title: "Logitech MX Master 3
Advanced Wireless Mouse"
LLM Response: "Yes, this product is
relevant to the query."
```

4.2 LLM Finetuning with Linear Layer

This method, shown in Figure 6 in the Appendix, involves appending a linear layer to the LLM's final hidden state output and fine-tuning this layer using a labeled dataset. The results from this method serve as our "Performance ceiling" or "Maximum achievable performance" and are reported in the last column of Table 1 & 2. The parameters of the linear layer are optimized by minimizing the binary cross-entropy loss, keeping the LLM's parameters fixed to preserve its pre-trained capabilities.

The LLM's output for the i-th query-product pair, denoted $\mathbf{h}_i$, is $\mathbf{h}_i = \theta_{LLM}(\mathrm{q}_i, \mathrm{p}_i)$ The linear layer applies a transformation to $\mathbf{h}_i$ to yield a relevance score $\hat{y}_i = \mathbf{W}\mathbf{h}_i + \mathbf{b}$. We use binary cross-entropy loss for training as $\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{CE}(y_i, \hat{y}_i)$, where y_i is the ground truth and $\hat{y}_i$ is the predicted probability of relevance. This enhanced approach, albeit slow and expensive for inference, effectively bridges the gap between general language understanding and specific task requirements, leading to marked performance improvements in relevance classification.

4.3 Cross-Encoder-Decoder for Reasoning Distillation

The LLM generates rationales using a carefully crafted prompt (see Appendix C) that emphasizes semantic connections and functional similarities between queries and product titles. Then reasoning prediction via decoder is used as an auxiliary task for training a cross-encoder model. This approach leverages the interpretability of LLM-generated reasoning to enhance the performance of a smaller, more efficient BERT cross-encoder model. We utilize the Mixtral 8X7B Instruct LLM from AWS Bedrock to generate reasoning for why a query-product pair is relevant or not. This process involves providing the LLM, θ_{LLM}, with a query-product-relevance triplet (q_i, p_i, y_i) with the designed prompt τ as $\tau(q_i, p_i, y_i)$ and asking it to generate a natural language explanation.

The input is formatted as: "[Query] is [Product Title] relevant? [Relevance (y_i)]. Explain why." The LLM response is a reasoning statement that explains the relevance decision.

Example:

Query: noise-cancelling headphones
Product Title: Bose QuietComfort 35 II
Relevance: Relevant
LLM Response: The product is relevant to the query because the Bose QuietComfort 35 II headphones are designed with noise-cancellation features, making them ideal for users looking to reduce ambient sound. The query specifically mentions 'noise-cancelling', which is a primary function of this product.

Our architecture, shown in Figure 1, consists of a transformer-based BERT cross-encoder and a T5 decoder. The cross-encoder handles the primary binary classification task, while the decoder generates reasoning based on the LLM's output. We use a standard BERT base encoder, 110M parameters with 12 layers and a 768-dimensional hidden state, a T5 base decoder for auto-regressive reasoning token prediction, and a binary classification head using a linear layer on the CLS token.

The query-product pair is fed into the BERT cross-encoder, producing a hidden state representation $\mathbf{h}_i$ as $\mathbf{h}_i = \text{BERT}(\text{q}_i, \text{p}_i)$. The CLS token representation $\mathbf{h}_{\text{CLS}}$ is used for generating predictions for binary classification as $\hat{y}_i = \sigma(\mathbf{W}\mathbf{h}_{\text{CLS}} + \mathbf{b})$. The overall token representations are fed into the T5 decoder to generate the reasoning sequence $\mathbf{r}_i$ as $\mathbf{r}_i = \text{Decoder}(\mathbf{h}_i)$ The total loss function combines the binary classification loss and the auxiliary reasoning generation loss:

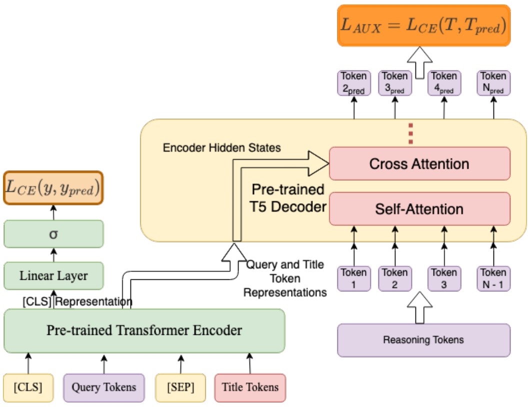


Figure 1: Cross-Encoder-Decoder for Reasoning Distillation

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \lambda \mathcal{L}_{\text{aux}}$$

The binary classification loss is defined as:

$$\mathcal{L}_{\text{classification}} = -\frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{CE}(y_i, \hat{y}_i)$$

The auxiliary task loss for reasoning generation is:

$$\mathcal{L}_{\text{aux}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \log P(r_{i,t} | r_{i,<t}, \mathbf{h}_i)$$

where y_i is the ground truth label, $\hat{y}_i$ is the predicted probability, $r_{i,t}$ is the reasoning token at time step t, $\mathbf{h}_i$ is the encoder hidden state from BERT, and λ is the weight for the auxiliary task loss. We set λ to 0.1 based on our early experiments.

4.4 Practical Modifications

Warmup of Decoder To stabilize learning, we initially train only the decoder on 50% of the data with a frozen BERT encoder. We then introduce a linear warmup phase for the encoder over the first 30% of training. During warmup, the loss focuses solely on auxiliary reasoning generation: $\mathcal{L}_{\text{warmup}} = \mathcal{L}_{\text{aux}}$. Post-warmup, the total loss combines binary classification and auxiliary task losses: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \lambda \mathcal{L}_{\text{aux}}$.

Ignore CLS Token for Decoder We experimented with including or excluding the CLS token, typically used for classification in transformers, in the decoder input. The hidden state representation $\mathbf{h}_i$ is adjusted accordingly based on this choice.

Embedding-based Reasoning As a compute-efficient alternative, this approach (Figure 4) compares embeddings of the reasoning to mean-pooled BERT cross-encoder representations. BGE-M3 (Chen et al., 2024) generates embeddings ($\mathbf{e}_{\text{reasoning}}$) for LLM-written reasonings. The cross-encoder's final layer output is mean-pooled ($\mathbf{h}_{\text{mean}}$), transformed ($\mathbf{h}_{\text{transformed}}$), and compared to $\mathbf{e}_{\text{reasoning}}$ using the loss function: $\mathcal{L}_{\text{aux}} = \|\mathbf{h}_{\text{transformed}} - \mathbf{e}_{\text{reasoning}}\|^2$.

5 Results and Ablations

Baseline We use pretrained BERT base model further trained on MS-MARCO query-passage-relevance dataset for search query relevance classification. We train this model using cross entropy loss to estimate $P(y_i|q_i, p_i)$, shown in Figure 5, we refer to this as **BERT** in our results tables.

Datasets and LLM Models: **(1).** 9 e-commerce regions for query-passage relevance: AU (English), ES (Spanish), BR (Portuguese), AE (English), FR (French), MX (Spanish), Saudi Arabia (Arabic), DE (German), and IN (English), each with 100K training and 10K test samples. All datasets used in our analysis are anonymized, aggregated, and do not represent production distribution. **(2).** Public ESCI dataset from Amazon for the US (English), JP (Japanese), and ES (Spanish) marketplaces, with 100K training and 10K test samples (Reddy et al., 2022). **(3).** GLUE benchmark and QADSM dataset for natural language inference and query-passage relevance classification (Wang et al., 2018; Liang et al., 2020). **(4).** LLaMA2-7B and Mistral-7B-v0.3 models for LLMs finetuning experiments (Touvron et al., 2023; Jiang et al., 2023). More details are provided in Appendix B.

5.1 Results

We refer to our method from Section 4.3 with Decoder warmup optimisation as **+ Reasoning (Ours)** and we also show the best results for LLM finetuning with linear layer from Section 4.2, taking best among the two LLMs (LLaMA2-7B and Mistral-7B-v0.3) as "Best of LLaMa2 and Mistral-7B". We run all our experiments 10 times each and report the mean and the 95% confidence interval in our tables. Under Appendix E we provide detailed results with results from both LLMs along with precision, recall and accuracy in Tables 5, 6, 7, 8. For details on the **reproducibility of our experiments and hyperparameter settings**, please refer to Appendix A.

Dataset	Samples (ZS/FS)	BERT	+ Reasoning (Ours)	Best of LLaMA2 and Mistral-7B
AU	10K	1x	**+2.21%**(±0.70%)	-0.06%(±0.58%)
	100K	1x	+1.21%(±1.20%)	+2.19%(±1.31%)
	ZS/FS			(-23.14%(±0.35%) / -15.53%(±0.69%))
ES	10K	1x	**+2.84%**(±1.11%)	-0.73%(±0.49%)
	100K	1x	+1.45%(±1.39%)	+3.98%(±1.16%)
	ZS/FS			(-28.01%(±0.37%) / -15.64%(±0.74%))
BR	10K	1x	**+0.37%**(±1.05%)	-1.69%(±0.82%)
	100K	1x	+1.14%(±1.21%)	+1.62%(±1.32%)
	ZS/FS			(-24.50%(±0.35%) / -18.30%(±0.58%))
AE	10K	1x	+1.45%(±0.94%)	+3.85%(±1.18%)
	100K	1x	+1.83%(±1.20%)	+3.13%(±1.31%)
	ZS/FS			(-29.77%(±0.35%) / -20.48%(±0.59%))
FR	10K	1x	**+2.11%**(±0.96%)	+1.85%(±0.84%)
	100K	1x	+2.16%(±1.24%)	+2.47%(±1.24%)
	ZS/FS			(-36.49%(±0.24%) / -18.12%(±0.56%))
MX	10K	1x	+0.32%(±0.84%)	+4.28%(±1.08%)
	100K	1x	+2.16%(±1.23%)	+3.64%(±1.34%)
	ZS/FS			(-27.87%(±0.36%) / -15.05%(±0.67%))
Arabia	10K	1x	**+1.56%**(±0.95%)	-0.45%(±0.72%)
	100K	1x	**+2.20%**(±1.21%)	+1.37%(±1.21%)
	ZS/FS			(-30.06%(±0.33%) / -19.68%(±0.55%))
DE	10K	1x	+2.14%(±0.85%)	+4.56%(±0.98%)
	100K	1x	+1.81%(±1.15%)	+4.97%(±1.26%)
	ZS/FS			(-24.70%(±0.37%) / -7.52%(±0.85%))
IN	10K	1x	**+1.76%**(±1.05%)	+0.61%(±0.94%)
	100K	1x	+0.87%(±1.24%)	+4.28%(±1.35%)
	ZS/FS			(-23.27%(±0.47%) / -12.37%(±0.67%))
		ESCI Dataset		
US	100K	1x	+2.55%(±1.29%)	+4.34%(±1.17%)
	ZS/FS			(-16.29%(±0.70%) / -9.61%(±0.94%))
JP	100K	1x	+1.53%(±1.09%)	+1.60%(±0.97%)
	ZS/FS			(-33.61%(±0.36%) / -25.55%(±0.61%))
ES	100K	1x	**+4.59%**(±1.29%)	+3.77%(±1.17%)
	ZS/FS			(-29.47%(±0.47%) / -15.45%(±0.82%))

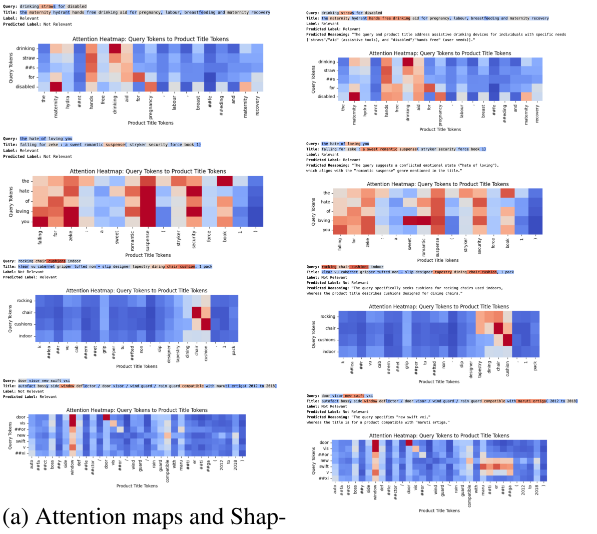
Table 1: Relative Performance Metrics Comparison Across Multilingual Datasets

Table 1 shows ROC-AUC metrics across 9 e-commerce datasets (upper part of table). Our reasoning method outperforms BERT baseline by 0.3-2.3 pp for 10K samples, surpassing Best LLM fine-tuned in 6/9 cases. At 100K samples, it remains competitive, outperforming in 1/9 cases. ZS/FS LLM performance is >20 pp lower, emphasizing finetuning necessity. For ESCI dataset (100K samples, bottom part of Table 1), our method shows 1.3-3.9 pp gains over BERT baseline across 3 regions surpassing even our "Performance ceiling" of 7B parameter finetuned LLM.

Table 2 shows results for 7 GLUE benchmark datasets. Our method outperforms the BERT baseline but not LLMs. The closest performance to LLMs is on QQP (0.9625 vs 0.97279) and MRPC (0.88105, a 0.09 improvement over BERT's 0.79548). LLMs' advantage on these challenging, low-resource tasks stems from extensive pretraining. However, for the QADSM query-passage relevance task, our method excels with an ROC-AUC of 0.91228, surpassing both BERT (0.71741) and the best LLM (0.87868) by 0.20 and 0.03 points

Dataset	BERT	+ Reasoning (Ours)	Best of LLaMA2 and Mistral-7B
QQP	0.9571(±0.006)	0.9625(±0.009)	0.9728(±0.004)
RTE	0.4754(±0.011)	0.5873(±0.007)	0.8867(±0.005)
MRPC	0.7955(±0.008)	0.8811(±0.003)	0.9486(±0.010)
QNLI	0.9467(±0.005)	0.9531(±0.012)	0.9887(±0.002)
Cola	0.5912(±0.009)	0.6130(±0.007)	0.9060(±0.011)
SST2	0.9488(±0.004)	0.9480(±0.006)	0.9907(±0.008)
QADSM	0.7174(±0.010)	**0.9123**(±0.005)	0.8787(±0.007)

Table 2: ROC-AUC of GLUE & QADSM benchmark



Query: drinking straws for disabled
Title: the maternity hydrant hands free drinking aid for pregnancy, labour, breastfeeding and maternity recovery
Label: Relevant
Predicted Label: Not Relevant

Query: the hate of loving you
Title: falling for zeke : a sweet romantic suspense(stryker security force book 1)
Label: Relevant
Predicted Label: Not Relevant

Query: rocking chair cushions indoor
Title: klear vu cabernet gripper tufted non – slip designer tapestry dining chair cushion, 1 pack
Label: Not Relevant
Predicted Label: Relevant

Query: door visor new swift vxi
Title: autofact bossy side window deflector / door visor / wind guard / rain guard compatible with maruti ertiga(2012 to 2018)
Label: Not Relevant
Predicted Label: Relevant

(a) Attention maps and Shapley visualization for BERT baseline model, demonstrating inconsistent focus on relevant tokens.

Query: drinking straws for disabled
Title: the maternity hydrant hands free drinking aid for pregnancy, labour, breastfeeding and maternity recovery
Label: Relevant
Predicted Label: Relevant
Predicted Reasoning: "The query and product title address assistive drinking devices for individuals with specific needs ["straws"/"aid" (assistive tools), and "disabled"/"hands free" (user needs)]."

Query: the hate of loving you
Title: falling for zeke : a sweet romantic suspense(stryker security force book 1)
Label: Relevant
Predicted Label: Relevant
Predicted Reasoning: "The query suggests a conflicted emotional state ("hate of loving"), which aligns with the "romantic suspense" genre mentioned in the title."

Query: rocking chair cushions indoor
Title: klear vu cabernet gripper tufted non – slip designer tapestry dining chair cushion, 1 pack
Label: Not Relevant
Predicted Label: Not Relevant
Predicted Reasoning: "The query specifically seeks cushions for rocking chairs used indoors, whereas the product title describes cushions designed for dining chairs."

Query: door visor new swift vxi
Title: autofact bossy side window deflector / door visor / wind guard / rain guard compatible with maruti ertiga(2012 to 2018)
Label: Not Relevant
Predicted Label: Not Relevant
Predicted Reasoning: "The query specifies "new swift vxi," whereas the title is for a product compatible with "maruti ertiga."

(b) Our model showcases improved token focus, attention on query-tokens and interpretability.

Figure 2: Attention maps, Shapley visualization, and generated reasoning for our proposed model, showcasing improved token relevance and interpretability.

respectively, demonstrating its effectiveness in e-commerce relevance classification.

5.2 Ablations

Figure 2 presents a comparative analysis of our proposed model against the BERT baseline for query-product relevance classification. The visualization includes attention maps and Shapley values, illustrating the models' focus on different tokens when determining relevance. Our model (Fig. 2b) demonstrates superior performance by consistently attending to semantically relevant tokens in both queries and product titles. In contrast, the baseline model (Fig. 2a) shows inconsistent attention patterns, often failing to identify the most relevant tokens for accurate classification. Additionally, we show that rationales generated from our cross-encoder-decoder are coherent and show deeper task understanding.

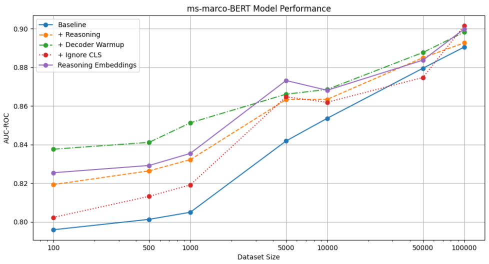


Figure 3: AUC-ROC vs Sample Size for various practical modifications.

Table 3: ROC-AUC scores for IN marketplace dataset with changing dataset sizes.

Samples	BERT	with Reasoning	+ Decoder Warmup	+ Ignore CLS	Reasoning Embeddings
100	1x	+2.94%	**+5.24%**	+0.80%	+3.71%
500	1x	+3.14%	**+4.98%**	+1.49%	+3.48%
1000	1x	+3.39%	**+5.76%**	+1.76%	+3.79%
5000	1x	+2.55%	+2.88%	+2.72%	**+3.72%**
10000	1x	+1.15%	**+1.75%**	+0.97%	+1.70%
50000	1x	+0.62%	**+0.93%**	-0.55%	+0.48%
100000	1x	+0.24%	+0.87%	**+1.24%**	+1.05%

Table 3 and Figure 3 compare our modifications from Sections 4.3 and 4.4 on the IN (English) dataset with varying training sample sizes. Our method with Decoder Warmup consistently performs best, providing up to 2 pp gain over reasoning alone. The compute-efficient Reasoning Embeddings approach outperforms the BERT baseline and nearly matches the full decoder method. Performance improves with sample size, but at a decreasing rate (e.g., 4.4 pp improvement from 100 to 10K samples, but only 2.9 pp from 10K to 100K). Our methods significantly outperform the baseline at lower sample sizes, demonstrating their efficacy in low-resource scenarios.

6 Conclusion

We developed a novel approach for enhancing relevance classification in e-commerce searches by integrating Large Language Models (LLMs) via knowledge distillation with a cost-efficient cross-encoder model. Our method leverages LLMs rationales during the training phase while only utilising the trained cross-encoder in production to achieve compute efficiency. Our experimental results confirm that this approach surpasses traditional models across various e-commerce datasets.

References

Sanjay Agrawal, Srujana Merugu, and Vivek Sembium. 2023a. Enhancing e-commerce product search through reinforcement learning-powered query reformulation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4488–4494.

Sanjay Agrawal, Vivek Sembium, and MS Ankith. 2023b. Kd-boost: Boosting real-time semantic matching in e-commerce with knowledge distillation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 131–141.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

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.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv*, abs/2004.01401.

Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2645–2652.

Sourab Mangrulkar, Ankith MS, and Vivek Sembium. 2022. Be3r: Bert based early-exit using expert routing. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3504–3512.

Michinari Momma, Chaosheng Dong, and Yetian Chen. 2022. Multi-objective ranking with directions of preferences.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. 2021. Zero-infinity: Breaking the GPU memory wall for extreme scale deep learning. *CoRR*, abs/2104.07857.

Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping queries dataset: A large-scale ESCI benchmark for improving product search.

V Sanh. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. 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, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Reproducibility and Hyperparameters

In this section, we describe the hyperparameters and training methodologies employed in our experiments, which utilize publicly available datasets and open-source models to ensure that our work can be independently verified and reproduced.

We conducted our experiments on the GLUE benchmark (Wang et al., 2018) and the ESCI dataset (Reddy et al., 2022), both of which are publicly available and widely used in the NLP community. For generating reasoning, we used the "Mixtral 8X7B Instruct" model provided by AWS Bedrock, which is available under the Apache 2.0 license. This model's open-source nature and the permissive licensing ensure that other researchers can use the same model for their work.

Hyperparameter	**Value**
Batch Size	32
Learning Rate	5e-5
Number of Epochs	8
Warmup Steps	500
Weight Decay on Decoder	0.001
Weight Decay on Encoder	0.0
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999
Gradient Clipping	1.0

Table 4: Hyperparameters used for training the models.

To ensure reproducibility, we provide the hyperparameters used in our experiments in Table 4, which were optimized through a series of preliminary trials. For our training methodology, we utilized PyTorch's Fully Sharded Data Parallel (FSDP) (Rajbhandari et al., 2021) to efficiently handle the large-scale models and datasets. FSDP allows us to shard model parameters, gradients, and optimizer states across data parallel workers, significantly reducing memory requirements and enabling the training of models on available multi-GPU hardware.

To generate reasoning with the "Mixtral 8X7B Instruct" model, we used the following prompt template, which was designed to elicit detailed explanations for relevance decisions:

```
Prompt:
"Given the following query-product pair,
is the product
relevant to the query?
Please provide reasoning for your answer.

Query: [Query]
Product Title: [Product Title]
Relevance: [Yes/No]

Reasoning:"
```

An example of a generated reasoning is as follows:

```
Example:
Query: noise-cancelling headphones
Product Title: Bose QuietComfort 35 II
Relevance: Yes

Reasoning: The Bose QuietComfort 35
II headphones are relevant to the query
because they are equipped
```

```
with advanced noise-cancelling technology,
which aligns with the user's search
for 'noise-cancelling headphones'.
This feature helps to
minimize ambient noise,
providing a quiet listening experience.
```

B Additional details on Datasets and LLM Models

For query-passage relevance, we use datasets from 9 e-commerce regions across with different languages - Australia (AU, English), Spain (ES, Spanish), Brazil (BR, Portuguese), United Arab Emirates (AE, English), France (FR, French), Mexico (MX, Spanish), Saudi Arabia (Arabic), Germany (DE, German), and India (IN, English). We also use the publicly available ESCI (E-commerce Search Corpus with Implicit user feedback) dataset (Reddy et al., 2022) from Amazon, which contains search sessions sampled from the Amazon Search Query Logs. The ESCI dataset is used to evaluate the performance across three marketplaces - United States (US, English), Japan (JP, Japanese), and Spain (ES, Spanish). Each of these datasets contains 100K training samples and 10K test samples. We also compute results by training on only 10K data points for our 9 e-commerce datasets.

For natural language inference, we use the GLUE benchmark (Wang et al., 2018) which includes 6 datasets that cover a range of natural language understanding tasks. We also use the QADSM (Question Answering Dataset on Search Media) dataset (Liang et al., 2020) for evaluating query-passage relevance classification. QADSM is a large-scale dataset designed for research on search relevance over e-commerce search media data like product titles and descriptions.

For our experiments involving large language models (LLMs), we use the LLaMA2-7B (Touvron et al., 2023) and Mistral-7B-v0.3 (Jiang et al., 2023) models. These LLMs are used to generate reasoning statements which are then used to train the smaller cross-encoder model through our proposed reasoning distillation approach. The performance of these LLMs on the zero-shot (ZS) and few-shot (FS) settings is also reported in the results tables for comparison.

C LLM Prompt for Rational Generation

In our approach, we utilize a Large Language Model (LLM) to generate rationals that inform the training of our smaller BERT model. The LLM is provided with a query, a product title, and their relevance label. It then generates a concise reasoning about the relevance between the query and the product title. The prompt used to guide the LLM in generating these rationales is as follows:

> Given a query, a product title and their relevance label, generate a concise reasoning (1-2 sentences) for their relevance. Focus on key semantic connections and functional similarities, rather than relying solely on exact word matches. Consider the following aspects:
> 1. Identify core functionality matches between the query and product title.
> 2. Recognize semantic relationships between different terms that serve similar purposes.
> 3. Align user needs across potentially different demographics or use cases.
> 4. Note shared purpose indicators or common structural elements in both query and title.
> 5. Connect conceptually related terms that may not be identical but serve similar functions.
> Emphasize how these connections demonstrate relevance despite potential differences in wording or target audiences.
> Your reasoning should highlight functional similarities and shared purposes that a classification model should learn to recognize when determining relevance between queries and product titles.
> Prioritize understanding context, functionality, and user needs to generate nuanced and accurate relevance determinations.
> Query: [Query]
> Product title: [Title]
> Actual Relevance label: [Relevance]
> Write concise reasoning for given relevance label.

This prompt is designed to guide the LLM in generating rationals that capture nuanced semantic relationships and functional similarities between queries and product titles, going beyond simple word matching. The generated rationals are then used to train our model, enhancing its ability to recognize complex relevance patterns in e-commerce scenarios.

D Figures detailing our methodology

In this section, we present architecture diagrams (Figure 4, 5, 6) which help show how various components of our approach work. Refer Section 4 for details on our methodology.

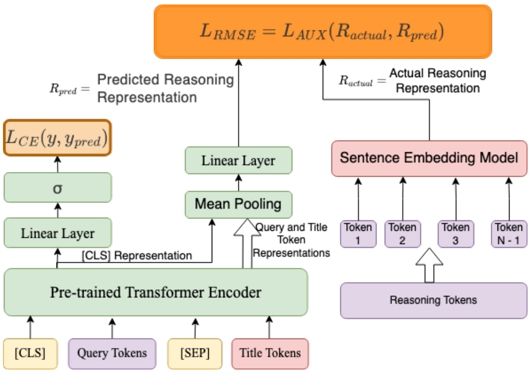

$L_{RMSE} = L_{AUX}(R_{actual}, R_{pred})$
R_{pred} = Predicted Reasoning Representation
R_{actual} = Actual Reasoning Representation
$L_{CE}(y, y_{pred})$
σ
Linear Layer
Linear Layer
Mean Pooling
Sentence Embedding Model
Query and Title Token Representations
[CLS] Representation
Token 1
Token 2
Token 3
Token N - 1
Pre-trained Transformer Encoder
Reasoning Tokens
[CLS]
Query Tokens
[SEP]
Title Tokens

Figure 4: Embedding-based Reasoning Process

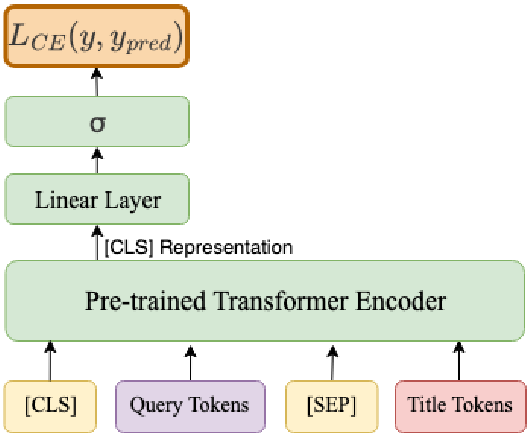


Figure 5: Architecture of Baseline method which trains an encoder for classification using cross-entropy loss.

E Detailed Results Tables

We present detailed results in this section which show results on both LLMs used (LLaMa2-7B and Mistral-7B-v0.3) as well as report precision, recall and accuracy as additional metrics for our experiments in Tables 5, 6, 7, 8. Tables in the main text are abridged versions of these tables.

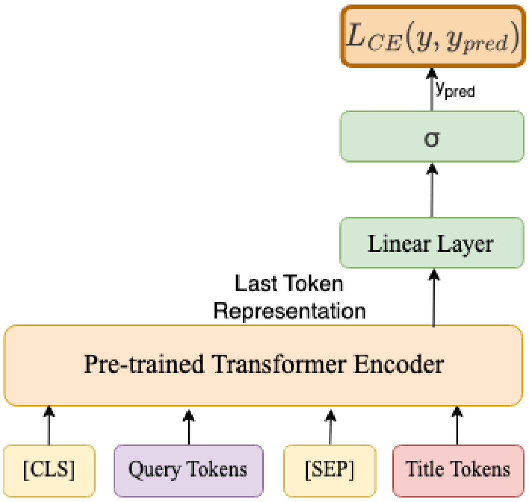


Figure 6: Architecture of LLM Fine-tuned with Linear Layer.

Marketplace	Model Type	Model/Approach	Samples	Accuracy	Recall	Precision	ROC-AUC
AU	LLM	LLaMA2-7B	10K	0.8670(±0.005)	0.9915(±0.003)	0.8678(±0.007)	0.8661(±0.004)
			100K	0.9008(±0.006)	0.9837(±0.002)	0.9054(±0.009)	0.9301(±0.008)
		Mistral-7B-V0.3	10K	0.8397(±0.011)	0.9993(±0.002)	0.8390(±0.006)	0.8361(±0.010)
			100K	0.9022(±0.004)	0.9837(±0.003)	0.9068(±0.007)	0.9354(±0.005)
	BERT	BERT (Baseline)	10K	0.8658(±0.008)	0.9437(±0.006)	0.9001(±0.004)	0.8666(±0.009)
			100K	0.8918(±0.007)	0.9800(±0.003)	0.9004(±0.005)	0.9153(±0.006)
		+ Reasoning (ours)	10K	0.8841(±0.005)	0.9620(±0.004)	0.9001(±0.008)	0.8857(±0.007)
			100K	0.9035(±0.006)	0.9970(±0.002)	0.9136(±0.009)	0.9264(±0.010)
ES	LLM	LLaMA2-7B	10K	0.8511(±0.007)	0.9743(±0.004)	0.8642(±0.006)	0.8026(±0.009)
			100K	0.8785(±0.005)	0.9796(±0.003)	0.8866(±0.008)	0.8997(±0.007)
		Mistral-7B-V0.3	10K	0.8554(±0.006)	0.9320(±0.005)	0.8983(±0.004)	0.7710(±0.010)
			100K	0.8883(±0.008)	0.9773(±0.002)	0.8977(±0.007)	0.8863(±0.006)
	BERT	BERT (Baseline)	10K	0.8378(±0.009)	0.8615(±0.007)	0.9000(±0.003)	0.8085(±0.005)
			100K	0.8592(±0.004)	0.9304(±0.006)	0.9000(±0.008)	0.8652(±0.007)
		+ Reasoning (ours)	10K	0.8483(±0.006)	0.8971(±0.005)	0.9177(±0.004)	0.8314(±0.008)
			100K	0.8724(±0.007)	0.9499(±0.003)	0.9107(±0.006)	0.8778(±0.009)
BR	LLM	LLaMA2-7B	10K	0.8507(±0.005)	0.9011(±0.007)	0.9182(±0.004)	0.8399(±0.006)
			100K	0.8966(±0.008)	0.9391(±0.003)	0.9369(±0.005)	0.9242(±0.007)
		Mistral-7B-V0.3	10K	0.7391(±0.009)	0.7761(±0.006)	0.8969(±0.004)	0.7128(±0.008)
			100K	0.9044(±0.005)	0.9488(±0.007)	0.9373(±0.003)	0.7516(±0.006)
	BERT	BERT (Baseline)	10K	0.8482(±0.007)	0.9348(±0.004)	0.9004(±0.006)	0.8543(±0.005)
			100K	0.8658(±0.006)	0.9695(±0.003)	0.9003(±0.008)	0.9094(±0.007)
		+ Reasoning (ours)	10K	0.8534(±0.004)	0.9532(±0.007)	0.9201(±0.005)	0.8575(±0.006)
			100K	0.8795(±0.008)	0.9820(±0.003)	0.9107(±0.006)	0.9198(±0.005)
AE	LLM	LLaMA2-7B	10K	0.8619(±0.006)	0.9589(±0.004)	0.8850(±0.007)	0.8486(±0.005)
			100K	0.9049(±0.007)	0.9599(±0.003)	0.9284(±0.006)	0.9305(±0.008)
		Mistral-7B-V0.3	10K	0.8784(±0.005)	0.9588(±0.008)	0.9016(±0.004)	0.8828(±0.007)
			100K	0.9138(±0.006)	0.9593(±0.003)	0.9386(±0.005)	0.9468(±0.004)
	BERT	BERT (Baseline)	10K	0.8404(±0.007)	0.9311(±0.005)	0.9001(±0.004)	0.8501(±0.006)
			100K	0.8824(±0.004)	0.9812(±0.007)	0.9005(±0.003)	0.9181(±0.005)
		+ Reasoning (ours)	10K	0.8522(±0.006)	0.9413(±0.004)	0.9003(±0.008)	0.8624(±0.007)
			100K	0.8937(±0.005)	1.0004(±0.003)	0.9107(±0.006)	0.9349(±0.004)
FR	LLM	LLaMA2-7B	10K	0.8360(±0.007)	0.9993(±0.002)	0.8360(±0.006)	0.8226(±0.008)
			100K	0.8826(±0.005)	0.9762(±0.004)	0.8929(±0.007)	0.9087(±0.003)
		Mistral-7B-V0.3	10K	0.8667(±0.006)	0.9335(±0.008)	0.9090(±0.004)	0.8484(±0.007)
			100K	0.8882(±0.004)	0.9677(±0.005)	0.9048(±0.006)	0.9108(±0.003)
	BERT	BERT (Baseline)	10K	0.8429(±0.007)	0.8954(±0.005)	0.9000(±0.004)	0.8330(±0.006)
			100K	0.8643(±0.005)	0.9482(±0.007)	0.9000(±0.003)	0.8888(±0.008)
		+ Reasoning (ours)	10K	0.8556(±0.006)	0.9135(±0.004)	0.9000(±0.007)	0.8506(±0.005)
			100K	0.8837(±0.004)	0.9590(±0.006)	0.9130(±0.003)	0.9080(±0.007)

Table 5: Performance Metrics of Different Models Across Various Marketplaces - Part 1

Marketplace	Model Type	Model/Approach	Samples	Accuracy	Recall	Precision	ROC-AUC
MX	LLM	LLaMA2-7B	10K	0.8432(±0.005)	0.9968(±0.003)	0.8435(±0.007)	0.8581(±0.004)
			100K	0.8843(±0.006)	0.9634(±0.002)	0.9040(±0.009)	0.9163(±0.008)
		Mistral-7B-V0.3	10K	0.8739(±0.011)	0.9857(±0.002)	0.8779(±0.006)	0.8703(±0.010)
			100K	0.9070(±0.004)	0.9505(±0.003)	0.9386(±0.007)	0.9261(±0.005)
	BERT	BERT (Baseline)	10K	0.8153(±0.008)	0.8908(±0.006)	0.9003(±0.004)	0.8346(±0.009)
			100K	0.8472(±0.007)	0.9529(±0.003)	0.9004(±0.005)	0.8936(±0.006)
		+ Reasoning (ours)	10K	0.8326(±0.005)	0.8929(±0.004)	0.9117(±0.008)	0.8373(±0.007)
			100K	0.8627(±0.006)	0.9682(±0.002)	0.9129(±0.009)	0.9129(±0.010)
Arabia	LLM	LLaMA2-7B	10K	0.8443(±0.007)	0.9891(±0.004)	0.8490(±0.006)	0.8341(±0.009)
			100K	0.8964(±0.005)	0.9675(±0.003)	0.9133(±0.008)	0.9226(±0.007)
		Mistral-7B-V0.3	10K	0.8633(±0.006)	0.9785(±0.005)	0.8729(±0.004)	0.7667(±0.010)
			100K	0.9096(±0.008)	0.9663(±0.002)	0.9282(±0.007)	0.9140(±0.006)
	BERT	BERT (Baseline)	10K	0.8501(±0.009)	0.9144(±0.007)	0.9000(±0.003)	0.8379(±0.005)
			100K	0.8875(±0.004)	0.9769(±0.006)	0.9001(±0.008)	0.9101(±0.007)
		+ Reasoning (ours)	10K	0.8674(±0.006)	0.9300(±0.005)	0.9000(±0.004)	0.8510(±0.008)
			100K	0.8983(±0.007)	0.9941(±0.003)	0.9195(±0.006)	0.9301(±0.009)
DE	LLM	LLaMA2-7B	10K	0.8432(±0.005)	0.9968(±0.007)	0.8435(±0.004)	0.8581(±0.006)
			100K	0.8843(±0.008)	0.9634(±0.003)	0.9040(±0.005)	0.9163(±0.007)
		Mistral-7B-V0.3	10K	0.8564(±0.009)	0.9848(±0.006)	0.8624(±0.004)	0.8242(±0.008)
			100K	0.8882(±0.005)	0.9677(±0.007)	0.9048(±0.003)	0.9108(±0.006)
	BERT	BERT (Baseline)	10K	0.8383(±0.007)	0.8736(±0.004)	0.9000(±0.006)	0.8207(±0.005)
			100K	0.8523(±0.006)	0.9302(±0.003)	0.9000(±0.008)	0.8729(±0.007)
		+ Reasoning (ours)	10K	0.8540(±0.004)	0.8888(±0.007)	0.9000(±0.005)	0.8383(±0.006)
			100K	0.8677(±0.008)	0.9438(±0.003)	0.9126(±0.006)	0.8887(±0.005)
IN	LLM	LLaMA2-7B	10K	0.8684(±0.007)	0.9774(±0.004)	0.8784(±0.006)	0.8588(±0.005)
			50K	0.8827(±0.006)	0.9822(±0.003)	0.8888(±0.008)	0.9117(±0.007)
			100K	0.8917(±0.005)	0.9804(±0.007)	0.8988(±0.004)	0.9286(±0.006)
		Mistral-7B-V0.3	10K	0.8671(±0.008)	0.9549(±0.005)	0.8930(±0.003)	0.7125(±0.009)
			50K	0.8851(±0.004)	0.9858(±0.006)	0.8885(±0.007)	0.8344(±0.005)
			100K	0.9004(±0.007)	0.9789(±0.003)	0.9087(±0.006)	0.8905(±0.008)
	BERT	BERT (Baseline)	10K	0.8427(±0.006)	0.9698(±0.004)	0.8563(±0.008)	0.8536(±0.007)
			50K	0.8792(±0.005)	0.9809(±0.007)	0.8727(±0.003)	0.8795(±0.006)
			100K	0.8902(±0.008)	0.9811(±0.003)	0.8992(±0.006)	0.8905(±0.005)
		+ Reasoning (ours)	10K	0.8795(±0.007)	0.9478(±0.004)	0.8950(±0.006)	0.8686(±0.005)
			50K	0.8840(±0.006)	0.9726(±0.008)	0.8979(±0.003)	0.8877(±0.007)
			100K	0.9013(±0.005)	0.9761(±0.003)	0.9125(±0.007)	0.8983(±0.006)

Table 6: Performance Metrics of Different Models Across Various Marketplaces - Part 2

Dataset	Method	Recall@0.7	Precision@0.7	ROC-AUC
qqp	Llama2-7B	0.8940(±0.006)	0.8410(±0.004)	0.9603(±0.008)
	Mistral-7B-v0.3	0.9083(±0.005)	0.8764(±0.007)	0.9728(±0.003)
	BERT (baseline)	0.7924(±0.009)	0.8788(±0.006)	0.9571(±0.004)
	+ Reasoning (ours)	0.8080(±0.007)	0.9000(±0.005)	0.9625(±0.010)
rte	Llama2-7B	0.7939(±0.008)	0.7761(±0.006)	0.8867(±0.004)
	Mistral-7B-v0.3	0.7634(±0.005)	0.7813(±0.009)	0.8690(±0.007)
	BERT (baseline)	0.0992(±0.003)	0.3514(±0.011)	0.4754(±0.006)
	+ Reasoning (ours)	0.1539(±0.007)	0.5790(±0.004)	0.5873(±0.008)
mrpc	Llama2-7B	0.9355(±0.006)	0.9063(±0.005)	0.9214(±0.009)
	Mistral-7B-v0.3	0.9391(±0.004)	0.9129(±0.007)	0.9486(±0.003)
	BERT (baseline)	0.8817(±0.008)	0.8066(±0.006)	0.7955(±0.005)
	+ Reasoning (ours)	0.8723(±0.007)	0.9023(±0.004)	0.8811(±0.010)
qadsm (en)	Llama2-7B	0.8174(±0.005)	0.7893(±0.008)	0.8511(±0.006)
	Mistral-7B-v0.3	0.8166(±0.007)	0.7965(±0.004)	0.8787(±0.009)
	BERT (baseline)	0.3398(±0.006)	0.7548(±0.005)	0.7174(±0.008)
	+ Reasoning (ours)	0.6462(±0.009)	0.9001(±0.003)	0.9123(±0.007)
qnli	Llama2-7B	0.9428(±0.004)	0.9563(±0.007)	0.9842(±0.005)
	Mistral-7B-v0.3	0.9453(±0.006)	0.9652(±0.003)	0.9887(±0.008)
	BERT (baseline)	0.7874(±0.009)	0.9295(±0.005)	0.9467(±0.004)
	+ Reasoning (ours)	0.7912(±0.007)	0.9457(±0.006)	0.9531(±0.010)
cola	Llama2-7B	0.9223(±0.005)	0.8614(±0.008)	0.8947(±0.006)
	Mistral-7B-v0.3	0.8988(±0.007)	0.9025(±0.004)	0.9060(±0.009)
	BERT (baseline)	0.7559(±0.006)	0.7325(±0.005)	0.5912(±0.008)
	+ Reasoning (ours)	0.7692(±0.009)	0.7780(±0.003)	0.6130(±0.007)
sst2	Llama2-7B	0.9685(±0.004)	0.9641(±0.007)	0.9907(±0.005)
	Mistral-7B-v0.3	0.9662(±0.006)	0.9684(±0.003)	0.9896(±0.008)
	BERT (baseline)	0.8536(±0.009)	0.9067(±0.005)	0.9488(±0.004)
	+ Reasoning (ours)	0.8419(±0.007)	0.9080(±0.006)	0.9480(±0.010)

Table 7: Detailed Performance Comparison on GLUE Benchmarks

Dataset	Method	Recall@0.7	Precision@0.7	ROC-AUC
US	Llama2-7B	0.9789(±0.005)	0.9145(±0.008)	0.8895(±0.006)
	Mistral-7B-v0.3	0.9757(±0.004)	0.9174(±0.007)	0.8920(±0.009)
	BERT (baseline)	0.9671(±0.006)	0.9099(±0.003)	0.8549(±0.010)
	+ Reasoning (ours)	0.9700(±0.007)	0.9162(±0.005)	0.8767(±0.004)
JP	Llama2-7B	0.9638(±0.009)	0.8817(±0.006)	0.8273(±0.005)
	Mistral-7B-v0.3	0.9549(±0.003)	0.8880(±0.008)	0.8383(±0.007)
	BERT (baseline)	0.9524(±0.005)	0.8836(±0.004)	0.8251(±0.009)
	+ Reasoning (ours)	0.9694(±0.008)	0.8807(±0.006)	0.8377(±0.003)
ES	Llama2-7B	0.9058(±0.007)	0.9086(±0.005)	0.8714(±0.008)
	Mistral-7B-v0.3	0.9429(±0.004)	0.8966(±0.009)	0.8846(±0.006)
	BERT (baseline)	0.9185(±0.006)	0.8863(±0.003)	0.8525(±0.007)
	+ Reasoning (ours)	0.9144(±0.005)	0.9109(±0.008)	0.8916(±0.004)

Table 8: Performance Comparison of Methods Across Different Regions for ESCI dataset