From a Large-Scale Teacher to Efficient Multimodal Students for Search Relevance Prediction

Tomasz Gniazdowski*
tgniazdowski@ebay.com
eBay
Tooploox
Wroclaw, Poland

Reza Momenifar*
rmomenifar@ebay.com
eBay
San Jose, USA

Nadia Vase
eBay
San Jose, USA

Shahram Khadivi
eBay
Aachen, Germany

Wiktor Walentynowicz
eBay
Tooploox
Wroclaw, Poland

Nicola Ueffing
eBay
Aachen, Germany

Jan Pawlowski
eBay
Tooploox
Wroclaw, Poland

Piotr Szymanski
eBay
Tooploox
Wroclaw, Poland

Abraham Bagherjeiran
eBay
San Jose, USA

Abstract

Relevance prediction between user queries and items is a fundamental component of e-commerce search systems. Traditionally, high-quality relevance labels are obtained through human judgment, which is reliable but expensive, slow, and difficult to scale. Recent advances in large language models (LLMs) enable automated relevance judgment at scale, achieving performance comparable to human annotators. However, such systems often rely on large commercial LLMs with complex prompts, resulting in high inference costs and limited throughput.

In this work, we propose a framework that bridges this gap by distilling a high-performing LLM-based relevance judgment system into efficient multimodal student models. We first curate a small, high-quality human-annotated dataset to serve as ground truth. Using this dataset, we develop a reliable *LLM-as-a-judge* pipeline through careful prompt engineering and iterative refinement. This pipeline is then used to generate large-scale relevance labels, which serve as training data for compact multimodal models. Our distilled models achieve performance close to the teacher system while significantly improving latency, cost, and scalability. This approach enables practical deployment of high-quality relevance prediction systems in production environments.

CCS Concepts

• **Information systems** → **Learning to rank**; *Relevance assessment*; Retrieval efficiency; Multimedia and multimodal retrieval.

Keywords

Multimodal Large Language Models, knowledge distillation, LLM as a Judge, search query relevance prediction

1 Introduction

Relevance prediction is a core task in information retrieval systems and plays a critical role in e-commerce search. Because it directly impacts ranking quality and user experience, accurately estimating the relevance between a user query and candidate items is essential for both evaluation and training of downstream models [Kabir et al. 2022; Thomas et al. 2023]. However, assessing the relevance of query-item pairs is fundamentally a commonsense reasoning task, requiring an understanding of user intent, product attributes, and contextual nuances. As a result, this task can be challenging even for experienced human judges, particularly when queries are ambiguous or item information is incomplete.

Traditionally, relevance labels are obtained through human annotation, where trained judges evaluate query-item pairs based on predefined guidelines. While reliable, this process is costly, time-consuming, and difficult to scale. Large production systems may spend millions annually yet generate only a limited number of labels, creating a bottleneck for model development and evaluation. Moreover, maintaining consistent and high-quality judgments across a wide range of queries and product categories requires significant domain expertise, which is difficult to ensure at scale.

Recent advances in large language models (LLMs) offer a scalable alternative. LLMs have demonstrated strong capabilities in reasoning and evaluation tasks, enabling the *LLM-as-a-judge* paradigm, where models generate relevance judgments given structured instructions and context [Faggioli et al. 2023]. Prior work has shown that commercial LLMs can produce relevance labels that align closely with human judgments [Soviero et al. 2024; Thomas et al. 2023]. However, directly deploying LLM-based relevance systems introduces a new challenge. High-performing solutions typically rely on large commercial models combined with long, carefully engineered prompts. While effective, these systems are expensive

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
3rd Workshop on Evaluating and Applying Recommender Systems with Large Language Models (EARL at CIKM '26), November 7–11, 2026, Rome, Italy
© Copyright held by the owner/author(s).

to run and have limited throughput, making them impractical for large-scale production use.

Taken together, these limitations reveal a fundamental trade-off between accuracy and scalability: human judgments are accurate but expensive and difficult to scale, while LLM-based judgments are scalable but remain computationally costly. To address this challenge, we propose a distillation-based framework for efficient relevance prediction. Prior works suggest that models trained via black-box knowledge distillation—where the token-level output distributions of a large, typically closed-source commercial teacher model are not accessible, can achieve comparable performance [Ho et al. 2023]. Our approach first constructs a small, high-quality human-annotated dataset, which is used to develop a reliable *LLM-as-a-judge* pipeline through prompt engineering and iterative refinement. This pipeline is then leveraged to generate large-scale labeled data, which serves as training data for compact multimodal student models. The resulting student models retain much of the teacher system's performance while significantly improving efficiency, enabling high-throughput, low-latency relevance prediction.

Our contributions can be summarized as follows:

- We propose a distillation-based framework that transfers knowledge from a high-performing, prompt-based *LLM-as-a-judge* system into efficient multimodal student models for search relevance prediction.
- We demonstrate that a commercial, closed-source LLM can be used as a reliable teacher to generate large-scale, high-quality relevance labels, enabling effective black-box distillation in search relevance prediction.
- We show that the distilled multimodal models achieve performance comparable to the teacher on human-annotated benchmarks, while significantly reducing inference cost and latency.
- We provide an empirical analysis of model stability, explanation quality, and efficiency trade-offs, demonstrating the suitability of the proposed approach for real-world deployment.

2 Related Work

In this section, we review prior work on applying LLMs to search relevance prediction, with a focus on *LLM-as-a-judge*, knowledge distillation, and synthetic data generation.

2.1 LLMs for Relevance Prediction and E-commerce Search

Large language models (LLMs) and vision-language models (VLMs) have been increasingly adopted in e-commerce applications, including product understanding, recommendation, and search. Prior work has explored adapting general-purpose LLMs to e-commerce domains through large-scale instruction tuning and domain-specific pretraining [Herold et al. 2025, 2024; Peng et al. 2024]. Other approaches leverage multimodal architectures to incorporate visual signals for product understanding [Nulli et al. 2026; Xue et al. 2023; Zhao et al. 2024].

In the context of search relevance prediction, both encoder-based and decoder-based models have been studied. Several works employ LLM-based teachers to supervise smaller models via distillation [Shang et al. 2025; Wang et al. 2024; Zhao et al. 2025], while others propose multi-stage pipelines combining supervised fine-tuning, reasoning, and reinforcement learning [Dong et al. 2025; Tang et al. 2025; Yang et al. 2026]. Despite these advances, most existing approaches rely either on expensive large models at inference time or require complex training pipelines, limiting their scalability in production environments.

2.2 *LLM-as-a-Judge* and Automated Evaluation

Recent work has introduced the *LLM-as-a-judge* paradigm, where large language models are used to evaluate outputs based on structured prompts and contextual instructions [Faggioli et al. 2023]. Empirical studies demonstrate that LLM-based evaluators can achieve strong agreement with human judgments across a variety of tasks, including dialogue evaluation, reasoning, and information retrieval [Soviero et al. 2024; Thomas et al. 2023].

At the same time, prior research highlights that the effectiveness of LLM-based evaluation is highly sensitive to prompt design and reasoning strategies [Brown et al. 2020; Wei et al. 2022]. Structured prompting, including detailed instructions, examples, and explicit rules, has been shown to significantly improve evaluation quality. However, deploying such systems in practice often requires complex, carefully engineered prompts and large models, resulting in high computational cost and limited throughput.

2.3 Knowledge Distillation and Synthetic Data Generation

Knowledge distillation has been widely used to transfer knowledge from large models to smaller, more efficient ones [Ho et al. 2023]. In the context of LLMs, previous works have studied both white-box distillation, where the student has access to the teacher's raw predictive distributions, and black-box distillation, where only the teacher's output text is available [Gu et al. 2024]. Several approaches leverage teacher-generated labels and explanations to improve student model performance [Agrawal et al. 2025; Zhao et al. 2025].

Closely related to our work, recent studies demonstrate the effectiveness of using LLMs to generate large-scale synthetic training data, enabling scalable supervision without human annotation. These approaches show that high-quality synthetic data can significantly improve downstream model performance while reducing reliance on costly human labeling [Abdin et al. 2024; Kang et al. 2025; Wang et al. 2025].

Unlike prior work that either relies on large LLMs at inference time or focuses on distillation without leveraging structured evaluation, our approach unifies *LLM-as-a-judge* evaluation, large-scale synthetic data generation, and multimodal knowledge distillation into a single framework. This enables the use of high-capacity commercial models as teachers while producing efficient models suitable for real-world deployment.

3 Methodology

We present the complete study pipeline below, including prompt engineering for the teacher model, automated construction of the student training dataset, and knowledge distillation from the teacher to the student models.

3.1 Ground Truth Curation

We define relevance prediction as the task of assigning a graded relevance label to a query-item pair, reflecting how well the item satisfies the user's intent. Following standard practices in information retrieval, relevance is commonly modeled using graded or ordinal labels [Kabir et al. 2022; Voorhees 2002]. In our setting, we adopt a four-level relevance scale: *Bad*, *Fair*, *Good*, and *Excellent*. While these labels provide a high-level abstraction, in practice each grade is governed by detailed annotation guidelines that specify how to handle factors such as exact matches, partial matches, substitutions, bundles, and other edge cases commonly encountered in e-commerce. To ground our system, we construct two human-annotated datasets of query-item pairs: a high-quality dataset annotated by three independent judges per example, and an additional dataset annotated by a single judge to increase coverage (hereafter referred to as the Test 1 and Test 2 datasets). The dataset annotated by three independent judges per example serves as a reference for calibrating the *LLM-as-a-judge* pipeline and evaluating its alignment with human judgments. The distribution of relevance labels for both datasets is provided in Appendix A.

3.2 Prompting Strategy

The effectiveness of the *LLM-as-a-judge* pipeline critically depends on prompt design, which defines how the model interprets the relevance evaluation task. We adopt a structured prompting strategy that encodes a consistent decision process, including task instructions, definitions of the relevance scale, explicit rules for handling common edge cases, and a set of curated few-shot examples. This design guides the model to reason systematically about query intent and item attributes, reducing ambiguity and improving consistency in predictions. The prompt is iteratively refined using feedback from ground truth annotations and systematic error analysis, where failure cases are analyzed and incorporated as additional rules or examples. This iterative process improves both the accuracy and robustness of the model's outputs. The resulting prompt enables the model to act as a reliable evaluator, which is essential for the subsequent *LLM-as-a-judge* pipeline used to generate large-scale training data for distillation.

3.3 LLM-as-a-Judge Pipeline

We formulate relevance prediction within the *LLM-as-a-judge* paradigm, where a large language model evaluates a query-item pair based on structured instructions and contextual information [Faggioli et al. 2023; Gu et al. 2026]. Given an input x and context C (prompt), the evaluation output is defined as:

$$\mathcal{E} \leftarrow \mathcal{P}_{\mathrm{LLM}}(x \oplus C),$$

where $\mathcal{E}$ denotes the predicted relevance label and an accompanying explanation.

In our setting, the input consists of a query q (user query text) and item metadata m, comprising a combination of textual and visual signals that characterize the item listing. The evaluation model M is instantiated as a prompted LLM and produces:

$$[r, e] \leftarrow M(p, q, m),$$

where p is a structured prompt, r is the relevance label, and e is a natural-language explanation.

In our framework, the model M serves as a **teacher model** within the *LLM-as-a-judge* pipeline. It is used to generate large-scale, high-quality supervision data for training downstream student models. For each query-item pair, the teacher produces both a relevance label and an explanation, forming training examples of the form (q, m, r, e).

Both components of the teacher output play an important role. The label provides the prediction target, while the explanation captures the reasoning process underlying the decision. This richer supervision signal improves the ability of the student model to approximate the teacher's behavior during distillation.

Empirically, we find that a *respond-then-explain* strategy—where the model first predicts the relevance label and then generates an explanation—yields more stable and reliable outputs compared to alternative reasoning strategies, while maintaining high efficiency.

We evaluate multiple commercial and open-source LLMs, including models from the GPT family. Among all evaluated models, `GPT-4.1` achieves the highest performance, reaching a multi-class relevance accuracy of approximately 83.7% on the human-annotated benchmark, demonstrating strong alignment with human judgments. Results for other models are provided in Appendix Table 9.

This strong alignment enables us to use `GPT-4.1` as a reliable teacher to generate large-scale, high-quality relevance labels and explanations, which are subsequently used to train efficient multimodal student models via knowledge distillation.

3.4 Distillation to Student Models

We train a compact multimodal autoregressive model, denoted M_{sft}, to predict relevance labels and optionally generate explanations:

$$[r_{\mathrm{sft}}, e_{\mathrm{sft}}] \leftarrow M_{\mathrm{sft}}(q, m).$$

Our goal is to approximate the behavior of the teacher model M (`GPT-4.1`), which produces predictions conditioned on a long prompt p:

$$[r, e] \leftarrow M(p, q, m).$$

To enable this distillation, we construct a large-scale dataset using the *LLM-as-a-judge* pipeline. For each example, we form:

$$X \leftarrow [p_{\mathrm{sft}}, q, m], \quad y \leftarrow [r, e],$$

where p_{sft} is a lightweight prompt suitable for efficient inference (see Appendix D), and (r, e) are generated by the teacher model. The resulting dataset $D = \{(X, y)\}$ captures the teacher's labeling behavior at scale.

We train M_{sft} via supervised finetuning starting from an instruction tuned base model M_{it}, selected from the `Gemma-3` [Team and DeepMind 2025] family of multimodal large language models. This choice is motivated by native support for text and image inputs, multilingual capabilities, and strong performance across benchmarks. Training is performed using the LLaMA-Factory framework [Zheng et al. 2024]. The student model learns to replicate both relevance labels and explanations, effectively distilling the teacher's behavior into a compact and efficient architecture suitable for production large-scale deployment.

For deployment, the student model can operate in a lightweight mode by generating only the relevance label
$r \in \{\texttt{bad}, \texttt{fair}, \texttt{good}, \texttt{excellent}\}$, significantly reducing inference

cost and latency. When required, explanations can also be generated by increasing the output length, without modifying the model architecture.

3.5 Datasets

The training and validation datasets are generated using the *LLM-as-a-judge* pipeline and are balanced across relevance classes and product categories to ensure diversity. In contrast, evaluation is conducted on human-annotated datasets: Test 1 (with 239 samples), which is annotated by three judges and used for prompt engineering for the teacher model, and Test 2 (with 233 samples) annotated by a single judge, completely independent from the prompt creation process. The relevance scores in all datasets are: *Bad, Fair, Good, Excellent*. The explanations are concise, typically consisting of a few sentences describing the key reasoning behind the assigned label.

Model selection is performed using validation data generated by the teacher model, while final evaluation relies on human-labeled data. This setup ensures scalability in training while maintaining reliable evaluation against high-quality human judgments. Additional details on data construction and preprocessing are provided in Appendix B. Dataset details are provided in Table 1.

4 Experiments

Initial finetuning configs are based on the supervised finetuning configurations of the State of the Art multimodal models: `DeepSeek-VL-7B` (*Stage 3*) [Lu et al. 2024], `Qwen-VL` (*Stage 3*) [Bai et al. 2023], `NVIDIA-Nemotron-Nano-V2-VL` (*Stage 4*) [Deshmukh et al. 2025], `Qwen-2.5-VL` (*SFT*) [Bai et al. 2025] and `Gemma-3` [Team and DeepMind 2025].

Next, we perform iterative hyperparameter tuning for all three `Gemma-3-IT` variants, namely the 4B, 12B, and 27B versions. For each model, we sequentially optimize a set of training hyperparameters. The hyperparameters considered, in the order in which they are tuned, are listed below. The best model is found based on the smallest loss value on Validation 2 dataset:

- number of epochs (from values $\in \{2, 3\}$),
- batch size (from values $\in \{256, 512, 1024, 2048, 4096\}$),
- weight decay (from values $\in \{0.05, 0.1\}$),
- learning rate (from values $\in \{1e^{-5}, 2e^{-5}, 5e^{-5}, 1e^{-4}\}$).

As in the `Gemma-3-IT` model SFT setup, we freeze the vision encoder in our models. The reason for that is to add regularization to the entire training process. As shown in Table 1, training and validation datasets are generated using the `GPT-4.1` teacher model. Test datasets, most of which contain edge cases, are annotated by human annotators. Freezing the vision encoder is intended to introduce a regularization factor to prevent the model from overfitting on synthetic and general data samples.

On this basis, the best sets of hyperparameters are found for finetuning the `Gemma-3-4B`, `Gemma-3-12B`, `Gemma-3-27B` models creating `Gemma-3-4B-IT-RLV`, `Gemma-3-12B-IT-RLV`, `Gemma-3-27B-IT-RLV` LLMs trained and evaluated on the datasets described above. The final best hyperparameter values are presented in Appendix Table 11.

Dataset	Description
Train (483,527)	Training dataset generated with `GPT-4.1`. Samples are balanced across class labels and categories.
Validation 1 (1,500)	Validation set generated with `GPT-4.1` and balanced across class labels. This split contains queries that also appear in the training set (of course for different items), resulting in broader category coverage and serving as an in-distribution validation benchmark.
Validation 2 (2,466)	Validation set generated with `GPT-4.1` and balanced across class labels and category names. Model selection is based on performance on this split. In contrast to Validation 1, it contains queries that do not appear in the training set, making it more challenging despite its narrower category coverage.
Test 1 (239)	Test set annotated independently by three human annotators. This split primarily contains edge cases and particularly challenging examples. This set is used in the process of optimizing prompt for the *LLM-as-a-judge* pipeline.
Test 2 (233)	Test set annotated by a single human annotator. This dataset is completely independent of the prompt engineering process for *LLM-as-a-judge*.

Table 1: Summary of the datasets used in our experiments, including split sizes, construction protocol, and annotation procedure. Validation 1 and Validation 2 differ in their degree of overlap with the training dataset queries, while Test 1 and Test 2 are human-annotated evaluation sets. Number of samples in each dataset is provided inside brackets.

4.1 Optimizer

AdamW optimizer [Loshchilov and Hutter 2019] is used for training with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Additionally model is trained using cosine learning rate scheduler [Loshchilov and Hutter 2017] with warm-up. It starts training at a very low learning rate value, increases it gradually to a peak over the warm-up fraction (in our setup equal to 0.1), and then decreases it following a cosine-shaped curve. This method helps stabilize very early training steps, when updates of the model weights are noisy, while the smooth cosine decay later allows smaller, gentler updates as the model converges.

4.2 Hardware

All training and evaluation experiments are conducted on the same hardware and software setup, using 8 NVIDIA A100 GPUs with CUDA 12.6. Only the throughput and performance experiments,

described in the Section 5.4, are conducted on different hardware—a single NVIDIA H100 GPU. Throughput is measured in FP8, while all models are otherwise run in FP16 precision. The implementation is based on the Llama Factory [Zheng et al. 2024], Pytorch 2.6.0 [Paszke et al. 2019] and Python 3.10, running on Ubuntu 22.04.

5 Results

The goal of this task is to achieve performance comparable to the teacher model on the Test 1 and Test 2 datasets, while significantly reducing inference cost and latency relative to the expensive `GPT-4.1` model. These datasets are annotated by three and one independent human judges, respectively, and contain many challenging examples that require substantial e-commerce expertise. Test 1 is useful for challenging edge cases but is not a fully untouched benchmark for the teacher pipeline; Test 2 is the cleanest estimate of independent human-evaluation generalization. The results for the student models—`Gemma-3-4B-IT-RLV`, `Gemma-3-12B-IT-RLV`, `Gemma-3-27B-IT-RLV` and encoder-based, text-only baseline model–finetuned on the same data as presented `Gemma-3-IT-RLV` models, without images and explanations, with classification manner `modernBERT-large` [Warner et al. 2025] are reported in Table 2. The task of the baseline model is to predict the same four-class relevance score as the student models, but without any explanation. The teacher model operates on lower-quality images (maximum edge length of 300 pixels), whereas the student models are trained and evaluated using higher-quality images (maximum edge length of 900 pixels). For completeness, results obtained using lower-quality images for the student models are provided in Appendix Table 13. Additionally, Appendix Table 12 analyzes the impact of training data size and diversity on model performance, Appendix Table 14 compares training with and without explanations, and Appendix Table 15 compares finetuning with and without images.

The ablation results show that both dataset scale and dataset diversity have a strong positive impact on model quality, as demonstrated in Appendix Section G.1. Training on the larger and more diverse dataset improves performance across almost all the models and evaluation datasets, with particularly clear gains for underrepresented classes and smaller models. Appendix Section G.2 shows that increasing the maximum image edge length from 300 to 900 pixels further improved results, especially for larger models and more difficult cases requiring subtle visual recognition. The results in Appendix Section G.3 indicate that adding explanations during training improved performance for most model and dataset variants, with the largest benefits observed on edge cases. Section G.4 in Appendix confirms that item images are an important input modality for reliable prediction.

Notably, image resolution has minimal impact on inference latency for `Gemma-3` models, as inputs are internally resized to fixed dimensions and presented as 256 visual tokens.

Overall, the student models achieve performance comparable to the teacher on the Test 1 dataset. On Test 2—an independent dataset that is not used during prompt design for the *LLM-as-a-judge* pipeline—the 12B model achieves even stronger results in terms of both accuracy and F1 score. Performance on validation datasets further indicates that the student models effectively replicate the teacher's behavior across most examples, demonstrating the effectiveness of the proposed knowledge distillation pipeline.

5.1 Per-class Results

The F1 metric values obtained per class for the finetuned models and the teacher's model are presented in Table 3. As expected, the middle classes—Fair and Good are more difficult for the models, they usually require more knowledge, and the differences between them are not as obvious as between the extreme classes—Bad and Excellent. This rule is visible for both the teacher model and the finetuned models. Similar conclusions can be drawn from the confusion matrices obtained for all the models on Test 1 and Test 2 datasets (Figures 1 and 2)—they are clearly diagonal, with weaker results obtained for the middle classes.

Based on the confusion matrices for Test 1 dataset, it can also be seen that most of the errors made by all analyzed models are equal to one step of difference between the prediction and the label value which is acceptable. The results for the Test 2 dataset are worse, but this was assumed, as it is a very demanding dataset, requiring extensive e-commerce knowledge and thorough analysis of all input signal features.

Analyzing the results of the models for the extreme and middle classes, it can also be noticed that the finetuned 12B model turned out to be more effective than the teacher model for both Test 1 and Test 2 datasets for middle classes. This is interesting because these samples are the ones that require more extensive e-commerce knowledge and very detailed analysis.

To sum up, when analyzing the per-class results, it can be said that the student models obtain similar results to the teacher model.



(a) `GPT-4.1` **(b) `Gemma-3-12B-IT-RLV` (our)**

Figure 1: Confusion matrices for teacher and `Gemma-3-12B-IT-RLV` models on Test 1 dataset.

5.2 Stability

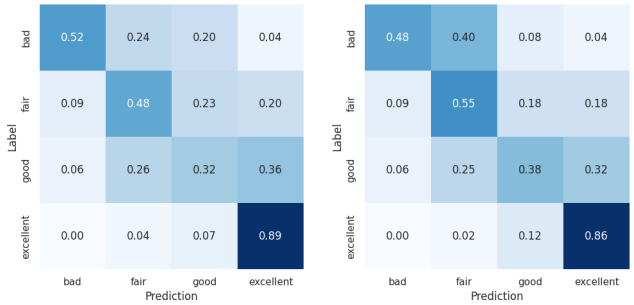
Results for stability experiment for `Gemma-3-4B-IT-RLV`, `Gemma-3-12B-IT-RLV`, `Gemma-3-27B-IT-RLV` models are presented in Table 4. For this experiment all finetuned models are evaluated 20 times on the Test 1 dataset to verify prediction stability. The table contains the mean and standard deviation values of the obtained results. With 239 samples in Test 1 dataset, one sample is 0.42% of the whole dataset. The variation between runs is very small: the

	GPT-4.1		modernBERT-large-RLV		Gemma-3-IT-RLV **(our)**					
	Teacher		Encoder Baseline		4B		12B		27B	
Dataset	acc [%]	F1 [%]	acc [%]	F1 [%]	acc [%]	F1 [%]	acc [%]	F1 [%]	acc [%]	F1 [%]
Validation 1	—	—	69.93	69.80	79.33	79.22	**82.33**	**82.39**	81.47	81.50
Validation 2	—	—	72.30	71.74	79.52	79.11	82.20	81.82	**83.33**	**82.96**
Test 1	**83.68**	**81.30**	74.06	61.29	80.34	76.19	81.59	77.78	81.59	78.71
Test 2	64.38	56.09	57.08	47.40	63.95	56.47	**65.24**	**57.56**	62.23	54.37

Table 2: Accuracy and F1 macro obtained for the finetuned models for search relevance classification for the Gemma-3-IT-RLV family models compared to the teacher GPT-4.1 model on all evaluation datasets.

Dataset	Model	Bad [%]	Fair [%]	Good [%]	Excellent [%]
Test 1	GPT-4.1	**100.00**	80.70	53.85	**90.63**
	Gemma-3-4B-IT-RLV **(our)**	92.31	76.00	47.62	88.82
	Gemma-3-12B-IT-RLV **(our)**	83.33	**84.62**	**54.35**	88.82
	Gemma-3-27B-IT-RLV **(our)**	92.31	82.35	51.16	89.02
Test 2	GPT-4.1	**57.78**	47.19	36.56	**82.85**
	Gemma-3-4B-IT-RLV **(our)**	53.33	50.00	41.18	81.39
	Gemma-3-12B-IT-RLV **(our)**	54.55	**51.61**	**41.67**	82.40
	Gemma-3-27B-IT-RLV **(our)**	51.16	47.83	36.89	81.58

Table 3: F1 metric values for search relevance classification for the Gemma-3-IT-RLV family models compared to the teacher GPT-4.1 model per class on Test 1 and Test 2 datasets.



(a) GPT-4.1 **(b) Gemma-3-12B-IT-RLV (our)**

Figure 2: Confusion matrices for teacher and Gemma-3-12B-IT-RLV models on Test 2 dataset.

Model	Accuracy
Gemma-3-4B-IT-RLV	(80.57 ± 0.21)%
Gemma-3-12B-IT-RLV	**(81.63 ± 0.29)%**
Gemma-3-27B-IT-RLV	(81.38 ± 0.25)%

Table 4: Mean accuracy and standard deviation values for all finetuned models for 20 independent model runs for search relevance classification.

standard deviation is lower than the share represented by a single example in the full Test 1 dataset. It can therefore be said that the predictions of all three models are stable across all the runs.

5.3 Explanation

The quality of the generated explanations for Gemma-3-4B-IT-RLV, Gemma-3-12B-IT-RLV, Gemma-3-27B-IT-RLV models are also examined. The chrf [Popović 2015] metric and the BERTScore F1 [Zhang* et al. 2020] are calculated for both lower-case predictions and labels and presented in Table 5. The chrf metric measures how similar the characters in the prediction and label are, while the BERTScore metric measures how similar the two texts are in meaning. High values of both metrics—especially BERTScore, which measures the semantic similarity of the texts, indicate that the quality of the generated explanations is high for all finetuned models. Please note here that the explanations are only in the validation datasets. The test datasets contain only relevance scores.

5.4 Throughput and Performance

Throughput and accuracy results for Gemma-3-12B-IT-RLV in FP8 and FP16 are reported in Table 6. All measurements are performed on a single NVIDIA H100 GPU, which supports FP8 computation. The results show that FP8-quantized models maintain comparable accuracy to their FP16 counterparts, while providing a substantial increase in maximum throughput. Given the high output quality of Gemma-3-12B-IT-RLV relative to Gemma-3-27B-IT-RLV (see Tables 2, 3, 4), together with its high maximum throughput and strong quantization performance (see Table 6), we select

Dataset	Model	chrf [%]	BS F1 [%]
Val 1	Gemma-3-4B-IT-RLV	67.15	93.78
	Gemma-3-12B-IT-RLV	67.97	93.89
	Gemma-3-27B-IT-RLV	**68.42**	**93.98**
Val 2	Gemma-3-4B-IT-RLV	66.22	93.45
	Gemma-3-12B-IT-RLV	67.48	93.70
	Gemma-3-27B-IT-RLV	**67.68**	**93.79**

Table 5: Models explanation quality—chrf and BERTScore F1 mean metric values (marked as BS F1 in Table) for all finetuned models from the Gemma-3-IT-RLV family calculated on Validation 1 (Val 1) and Validation 2 (Val 2) datasets.

Gemma-3-12B-IT-RLV-FP8 as the model that best represents the Pareto frontier sweet spot. In particular, it offers the best trade-off between metric performance and throughput, achieving strong evaluation results while maintaining very high serving efficiency. Appendix Table 16 provides a comparison of the cost of 1 million input tokens and 1 million output tokens for the teacher and Gemma-3-12B-IT-RLV-FP8 models. The results indicate that the student models are substantially more cost-effective in both input- and output-token costs relative to GPT-4.1.

Dataset	Gemma-3-12B-IT-RLV	
	FP16	FP8
Validation 1 [%]	82.33	82.00 [-0.33]
Validation 2 [%]	82.20	81.87 [-0.33]
Test 1 [%]	81.59	81.17 [-0.42]
Test 2 [%]	65.24	65.67 [+0.43]
TP w/o explan. [req/s]	194.52	271.54 [+77.02]
TP w/ explan. [req/s]	49.57	55.98 [+6.41]

Table 6: Accuracy [%] and throughput for the Gemma-3-12B-IT-RLV models quantized to FP16 and FP8. Throughput (marked in Table as TP) is reported in requests per second [req/s]. Experiments are conducted on a single NVIDIA H100 GPU.

6 Conclusion

In this work, we presented a scalable framework for search relevance prediction that bridges the gap between high-quality but expensive human and LLM-based judgment systems and efficient production-ready models. By leveraging a carefully designed *LLM-as-a-judge* pipeline, we generate large-scale, high-quality relevance labels that enable effective knowledge distillation into compact multimodal student models. Experimental results demonstrate that the distilled Gemma-3-IT-RLV models achieve performance comparable to the GPT-4.1 teacher on human-annotated benchmarks, while significantly reducing inference cost and latency.

Gemma-3-12B-IT-RLV-FP8 is the best Pareto point because it is near-teacher quality, stronger than the encoder baseline, competitive with or better than the 27B model variant on the independent human Test 2 dataset, and much faster and cheaper in FP8. The student models also exhibit strong generalization, stability across runs, and high-quality explanation generation, confirming that they successfully capture the behavior of the teacher system. Overall, our results show that it is possible to retain the accuracy of large, prompt-based LLM systems while enabling efficient, high-throughput relevance prediction suitable for real-world deployment. This work highlights the potential of combining *LLM-as-a-judge* systems with knowledge distillation to build scalable and practical relevance modeling pipelines for large-scale search applications.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905* (2024).

Sanjay Agrawal, Faizan Ahemad, and Vivek Varadarajan Sembium. 2025. Rationale-guided distillation for e-commerce relevance classification: Bridging large language models and lightweight cross-encoders. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*. 136–148.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631* (2025).

Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language Models are Few-Shot Learners. *NeurIPS* (2020).

Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Guo Chen, et al. 2025. Nvidia nemotron nano v2 vl. *arXiv preprint arXiv:2511.03929* (2025).

Chenhe Dong, Shaowei Yao, Pengkun Jiao, Jianhui Yang, Yiming Jin, Zerui Huang, Xiaojiang Zhou, Dan Ou, Haihong Tang, and Bo Zheng. 2025. TaoSR1: The thinking model for e-commerce relevance search. *arXiv preprint arXiv:2508.12365* (2025).

Guglielmo Faggioli, Laura Dietz, Charles Clarke, et al. 2023. Perspectives on Large Language Models for Relevance Judgment. In *SIGIR ICTIR*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. 2026. A survey on LLM-as-a-judge. *The Innovation* (2026), 101253. doi:10.1016/j.xinn.2025.101253

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*.

Christian Herold, Michael Kozielski, Tala Bazazo, Pavel Petrushkov, Yannick Versley, Seyyed Hadi Hashemi, Patrycja Cieplicka, Dominika Basaj, and Shahram Khadivi. 2025. Domain adaptation of foundation llms for e-commerce. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*. 1039–1049.

Christian Herold, Michael Kozielski, Leonid Ekimov, Pavel Petrushkov, Pierre-Yves Vandenbussche, and Shahram Khadivi. 2024. LiLiuM: eBay's Large Language Models for e-commerce. *arXiv preprint arXiv:2406.12023* (2024).

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*. 14852–14882.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. FastText.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651* (2016).

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Mirella Lapata, Phil Blunsom, and Alexander Koller (Eds.). Association for Computational Linguistics, Valencia, Spain, 427–431. https://aclanthology.org/E17-2068/

Md Ahsanul Kabir et al. 2022. Ordinal Regression for E-commerce Query Similarity Prediction. *arXiv preprint arXiv:2203.06591* (2022).

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. 2025. Distilling LLM Agent into Small Models with Retrieval and Code Tools. In *Advances in Neural Information Processing Systems*. https://mlanthology.org/neurips/2025/kang2025neurips-distilling/

Lambda. 2026. AI Cloud Pricing | GPU Compute & AI Infrastructure. https://lambda.ai/pricing Accessed: 2026-05-13.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Skq89Scxx

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. 2024. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525* (2024).

Matteo Nulli, Orshulevich Vladimir, Tala Bazazo, Christian Herold, Michael Kozielski, Marcin Mazur, Szymon Tuzel, Cees Snoek, Seyyed Hadi Hashemi, Omar Javed, et al. 2026. Adapting Vision-Language Models for E-commerce Understanding at Scale. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*. 496–512.

OpenAI. 2026. Pricing. https://developers.openai.com/api/docs/pricing Accessed: 2026-05-13.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

Bo Peng, Xinyi Ling, Ziru Chen, Huan Sun, and Xia Ning. 2024. eCeLLM: Generalizing Large Language Models for E-commerce from Large-scale, High-quality Instruction Data. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 40215–40257. https://proceedings.mlr.press/v235/peng24c.html

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina (Eds.). Association for Computational Linguistics, Lisbon, Portugal, 392–395. doi:10.18653/v1/W15-3049

Runpod. 2026. Pricing. https://www.runpod.io/pricing Accessed: 2026-05-13.

SeatGeek. 2023. thefuzz: Fuzzy String Matching in Python. https://github.com/seatgeek/thefuzz. Version 0.22.1, GitHub repository.

Hongwei Shang, Nguyen Vo, Nitin Yadav, Tian Zhang, Ajit Puthenputhussery, Xunfan Cai, Shuyi Chen, Prijith Chandran, and Changsung Kang. 2025. Knowledge distillation for enhancing walmart e-commerce search relevance using large language models. In *Companion Proceedings of the ACM on Web Conference 2025*. 449–457.

Beatriz Soviero, Daniel Kuhn, and Alexandre Salle. 2024. ChatGPT Goes Shopping: LLMs Can Predict Relevance in E-commerce Search. In *ECIR*.

Tian Tang, Zhixing Tian, Zhenyu Zhu, Chenyang Wang, Haiqing Hu, Guoyu Tang, Lin Liu, and Sulong Xu. 2025. Lref: A novel llm-based relevance framework for e-commerce search. In *Companion Proceedings of the ACM on Web Conference 2025*. 468–475.

Gemma Team and Google DeepMind. 2025. Gemma 3 Technical Report. https://arxiv.org/abs/2503.19786. arXiv:2503.19786 [cs.CL]

Qwen Team. 2025. Qwen3 Technical Report. https://arxiv.org/abs/2505.09388. arXiv:2505.09388 [cs.CL]

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large Language Models Can Accurately Predict Searcher Preferences. *arXiv preprint arXiv:2309.10621* (2023).

Ellen M. Voorhees. 2002. The Philosophy of Information Retrieval Evaluation. *CLEF* (2002).

Chengyu Wang, Junbing Yan, Yuanhao Yue, and Jun Huang. 2025. Distilqwen2. 5: Industrial practices of training distilled open lightweight language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*. 32–42.

Han Wang, Mukuntha Narayanan S, Onur Gungor, Yu Xu, Krishna Kamath, Rakesh Chalasani, Kurchi Subhra Hazra, and Jinfeng Rao. 2024. Improving Pinterest Search Relevance Using Large Language Models. In *Proceedings of 3rd International Workshop on Industrial Recommendation Systems (at CIKM 2024) (CIKM '24)*. ACM, New York, NY, USA, 6. doi:XXXXXXX.XXXXXXX

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 2526–2547. doi:10.18653/v1/2025.acl-long.127

Jason Wei, Xuezhi Wang, and Dale Schuurmans. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *NeurIPS* (2022).

Wei Xue, Zongyi Guo, Baoliang Cui, Zheng Xing, Xiaoyi Zeng, Xiufei Wang, Shuhui Wu, and Weiming Lu. 2023. Pumgpt: A large vision-language model for product understanding. *arXiv preprint arXiv:2308.09568* (2023).

Jianhui Yang, Yiming Jin, Pengkun Jiao, Chenhe Dong, Zerui Huang, Shaowei Yao, Xiaojiang Zhou, Dan Ou, and Haihong Tang. 2026. TaoSR-AGRL: Adaptive Guided Reinforcement Learning Framework for E-commerce Search Relevance. In *Proceedings of the ACM Web Conference 2026* (United Arab Emirates) *(WWW '26)*. Association for Computing Machinery, New York, NY, USA, 7955–7966. doi:10.1145/3774904.3792799

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkeHuCVFDr

Gang Zhao, Ximing Zhang, Chenji Lu, Hui Zhao, Tianshu Wu, Pengjie Wang, Jian Xu, and Bo Zheng. 2025. Explainable LLM-driven Multi-dimensional Distillation for E-Commerce Relevance Learning. In *Companion Proceedings of the ACM on Web Conference 2025*. 631–640.

Xiangyu Zhao, Yuehan Zhang, Wenlong Zhang, and Xiao-Ming Wu. 2024. UniFashion: A Unified Vision-Language Model for Multimodal Fashion Retrieval and Generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 1490–1507. doi:10.18653/v1/2024.emnlp-main.89

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, Bangkok, Thailand. http://arxiv.org/abs/2403.13372

A Test Dataset Distributions

We construct two human-annotated datasets to evaluate the alignment between LLM-generated judgments and human relevance assessments. Both datasets use the same four-level relevance scale: *Bad*, *Fair*, *Good*, and *Excellent*.

A.1 Test 1 Dataset

The first dataset consists of 239 query-item pairs, each independently annotated by three human judges. The final label for each pair is determined based on the annotated judgments, allowing for more robust and reliable ground truth through agreement across multiple annotators. Table 7 shows the distribution of relevance labels for this dataset.

Label	Bad	Fair	Good	Excellent
Count	6	26	43	164

Table 7: Label distribution for the multi-annotator dataset (239 samples, 3 judges per sample).

A.2 Test 2 Dataset

The second dataset consists of 233 query-item pairs, each annotated by a single human judge. While this dataset provides additional coverage, it may exhibit higher variability due to the absence of inter-annotator agreement. Table 8 shows the distribution of relevance labels for this dataset.

Label	Bad	Fair	Good	Excellent
Count	25	44	53	111

Table 8: Label distribution for the single-annotator dataset (233 samples, 1 judge per sample).

B Data Details

Records are preprocessed by removing duplicates, dropping entries with missing mandatory fields (user query, model prediction), filtering to English-language texts only using FastText language detection model [Joulin et al. 2016, 2017], and enforcing minimum/-maximum character length thresholds.

After a detailed analysis of the problem, we identify a set of special-case examples that are excluded from the downsampling procedure and retained in the dataset. Specifically, these are data points containing predefined keywords and are therefore preserved in the training or validation splits. The presence of such keywords suggests that a sample may correspond to a challenging edge case that is important for the task, motivating its inclusion despite downsampling. The rest of the data is downsampled. To obtain a relatively balanced dataset, weighted sampling is used, in which the final weight value w is determined from the formula: $w \leftarrow w_{label} + 0.1 * w_{category}$. The sampling weights w_{label} and $w_{category}$ are calculated for respectively: label generated with `GPT-4.1` (relevance score) and item category with `compute_sample_weight` function from Scikit-learn library [Pedregosa et al. 2011].

To prevent data leakage, training candidates are also checked against all evaluation sets using both exact matching and fuzzy logic matching using thefuzz Python library [SeatGeek 2023].

The entire process described above results in the creation of Training and Validation 2 datasets. Because data creation is a long and iterative process, we also have Validation 1 data, which is created in a similar way at initial stages of the study. These datasets are not concatenated because Validation 1 contains user queries that are present in the training data (only for other items). Validation 2 on the other hand does not contain queries that are present in the training dataset—later in the project, we have more data samples available. As a result, Validation 1 dataset has 1500 balanced samples, it contains queries that are in training data, but because of that, the categories of the datapoints are more various. On the other hand Validation 2 dataset contains 2466 balanced samples and contains queries that are not present in the training dataset. Because of that categories of the datapoints are less diverse (item categories are inextricably linked to user queries). However, the created dataset is more difficult, as the queries from Validation 2 that determine the user's needs, do not appear in the Training dataset. For this reason, it is decided to separate both sets and provide two metrics for each of them. Final Training dataset contains 483,527 samples.

The resulting datasets span a broad and heterogeneous set of item categories. The most common meta-categories include: collectibles, fashion, electronics, and vehicle parts/accessories. Additional substantial groups correspond to sports equipment, toys, home goods, media, health and beauty, jewellery, and business/industrial supplies, indicating that the dataset captures a diverse cross-section of marketplace inventory.

The Test datasets are also created iteratively and, unlike the validation data, are annotated by human annotators. The cost of obtaining such annotations is enormous, with 239 samples obtained for Test 1 dataset and 233 samples for Test 2 dataset.

C Results for Other Large Language Models

Table 9 summarizes the results for the commercial LLMs: `GPT-5`, `GPT-4.1`, `GPT-4o`, `GPT-4.1-mini`, and `GPT-4.1-nano` and open-source LLMs: `Gemma-3-27B-IT` and `Qwen3-VL-32B-Instruct` [Bai et al. 2025] prompted with the prepared prompt described in Section 3.2.

Weights	Model	Test 1 [%]
closed-source	`GPT-5`	80.33
	`GPT-4.1`	**83.68**
	`GPT-4o`	79.92
	`GPT-4.1-mini`	74.06
	`GPT-4.1-nano`	55.65
open-source	`Gemma-3-27B-IT`	68.20
	`Qwen3-VL-32B-Instruct`	72.38

Table 9: Accuracy scores [%] on the Test 1 dataset for the commercial and open-source LLMs. All models are prompted using the prompt described in Section 3.2.

D Supervised Finetuning Prompt

The prompt used to finetune all `Gemma-3-IT-RLV` models is shown below. In the production setting, this prompt can be cached, and only the input item data and user query are appended.

SFT prompt

Based on the given query, item metadata, determine how relevant the item is to the query. Return a value specifying the level of relevance from the list: ['bad', 'fair', 'good', 'excellent'] and explanation.
<item metadata>
<user query>

E `Gemma-3-12B-IT-RLV` vs `Gemma-3-27B-IT-RLV` Explanation Quality

Furthermore chrf and BERTScore metrics are calculated between the 12B and 27B models, to determine the degree of explanation quality degradation for the smaller model. Lower-case predictions of `Gemma-3-27B-IT-RLV` are set as labels and lower-case predictions of `Gemma-3-12B-IT-RLV` as predictions. For this task, `Qwen3-32B` [Team 2025] is also used (a totally different model is used on purpose to avoid introducing bias) as a *LLM-as-a-judge* to assess whether the given texts are same, similar, or completely different in meaning. The LLM prompt for this task can be found below.

Based on the analysis of the results comparing the explanation quality of 12B and 27B models (Table 10), the quality of the explanations provided by the smaller model is comparable to those predicted by the larger one. Most of the explanations are assessed by *LLM-as-a-judge* as the same or similar (2%—4% of the explanations across all the evaluation datasets are judged as different). The similarity of both explanations across all datasets is also evidenced

by the high mean BERTScore F1 metric values (≈ 95% across all datasets).

Prompt for an *LLM-as-a-judge* text similarity task

Task: Compare Text A and Text B and output only one word: same, similar, or different.
Criteria:
* same: identical meaning or paraphrase; differences are stylistic/trivial,
* similar: same core idea/intent with minor additions/omissions,
* different: conflicting meanings, topics, claims, or intents.
Consider negation, scope, time, quantities, and entities. Ignore wording, order, and formatting. Do not return explanation.
Text A: <text a>
Text B: <text b>

Dataset	chrf	BS F1	LLM judge		
			same	similar	different
Val 1	75.07	95.25	875 (58.3%)	590 (39.3%)	35 (2.3%)
Val 2	74.81	95.06	1532 (62.1%)	852 (34.6%)	82 (3.3%)
Test 1	75.27	95.17	142 (59.4%)	91 (38.1%)	6 (2.5%)
Test 2	73.27	94.79	125 (53.7%)	99 (42.5%)	9 (3.9%)

Table 10: Models explanation quality—chrf, BERTScore F1 mean metric (marked in Table as BS F1) values and *LLM-as-a-judge* results. Here the lower-case `Gemma-3-27B-IT-RLV` predictions are set as labels and lower-case predictions of `Gemma-3-12B-IT-RLV` as predictions on Validation 1 and Validation 2 datasets (marked as Val 1 and Val 2 in Table) and Test 1 and Test 2 datasets.

F The Best Hyperparameter Sets

The best found sets of hyperparameters for all finetuned models are presented in Table 11.

Hyperparameter	`Gemma-3-IT-RLV`		
	4B	12B	27B
Learning rate	$2e^{-5}$	$2e^{-5}$	$2e^{-5}$
LR scheduler	cosine	cosine	cosine
Weight decay	0.1	0.1	0.1
Gradient clip	1.0	1.0	1.0
Optimizer	AdamW (0.9, 0.999)	AdamW (0.9, 0.999)	AdamW (0.9, 0.999)
Warm-up fraction	0.1	0.1	0.1
Epochs	2	2	2
Batch size	512	1024	2048
Sequence length	1024	1024	1024
Freeze vision enc	Yes	Yes	Yes
Freeze mm projector	No	No	No

Table 11: The best hyperparameter sets found for all analyzed models. Hyperparameters for AdamW optimizer β_1 and β_2 are presented in the brackets.

G Ablations

The influence of the number and variety of samples and image sizes in the training set is investigated. The results are presented in sections G.1 and G.2

G.1 Number of Data Samples

Because the datapoints are collected iteratively from the production processes, the impact of number of samples on the final model quality can be easily compared. Table 12 shows the impact of sample count and sample diversity on the final model quality. Training set version 2 (v2) is described in Table 1. Training dataset version 1 (v1) is created from less diverse data, with 52,456 samples. The following results are obtained for all models and the maximum edge length of the images in both datasets is 300 pixels for fair comparison. New hyperparameter sets are also found for all models trained on the smaller dataset.

Using approximately nine times more training data in Train 2 dataset than in Train 1 dataset, together with the inclusion of a broader range of special cases and diverse examples across more categories, led to a consistent improvement in the performance of all the trained models across all evaluation datasets. The only exception is `Gemma-3-12B-IT-RLV` on Test 2 dataset. Models trained on the larger dataset (v2) improve accuracy by 3.35 p.p., 4.19 p.p. and 1.26 p.p. over the models trained on dataset v1 on Test 1 dataset. The difference in the prediction quality for Test 1 dataset of the models is especially visible for the F1 macro metric values, which take into account the prediction quality for underrepresented classes (Test 1 dataset is unbalanced). As expected, using a larger number of more diverse examples improves the model's performance on underrepresented classes. Smaller models (4B and 12B) generally benefit more from v2 dataset than 27B model. This is probably because the 27B model starts with broader prior knowledge. It is also worth noting that the 12B and 27B models trained on v2 dataset achieved similar results across all datasets, so the difference resulting from the different initial level of the models knowledge is minimized after training the models on larger dataset. Interestingly, the transition from Train dataset 1 to Train dataset 2 often yields larger gains than increasing model size, for example when comparing the 4B v2 model against the 12B v1 model, and the 12B v2 model against the 27B v1 model.

G.2 Image Size

Then the impact of image size is compared on the final model quality. The results are presented in Table 13. The results from the previous experiment for all the models trained on Training 2 dataset with images with a maximum edge length of 300 pixels (low quality) are compared with the best models trained on the same examples with images having maximum edge length of 900 pixels (high quality).

On Test 1 dataset usage of bigger images (900 pixels instead of 300 pixels) improves accuracy by 0.84 p.p., 1.67 p.p. and 2.09 p.p. for (respectively) 4B, 12B and 27B models. The bigger the model is, the

Model name	Metric	Gemma-3-4B-IT-RLV		Gemma-3-12B-IT-RLV		Gemma-3-27B-IT-RLV	
Train dataset version		v1	v2	v1	v2	v1	v2
Validation 1	acc [%]	78.00	79.00 [+1.00↑]	80.13	81.27 [+1.14↑]	80.67	81.20 [+0.53↑]
Test 1	acc [%]	76.15	79.50 [+3.35↑]	75.73	79.92 [+4.19↑]	78.24	79.50 [+1.26↑]
	F1 [%]	68.92	71.03 [+2.11↑]	64.20	76.96 [+12.76↑]	69.85	76.82 [+6.97↑]
Test 2	acc [%]	61.80	61.80 [-]	65.24	62.66 [-2.58↓]	63.09	64.38 [+1.29↑]

Table 12: The influence of the dataset version (number of datapoints and their diversity) on the search relevance classification quality.

Model name	Metric	Gemma-3-4B-IT-RLV		Gemma-3-12B-IT-RLV		Gemma-3-27B-IT-RLV	
Max image edge length		300 px	900 px	300 px	900 px	300 px	900 px
Validation 1	acc [%]	79.00	79.33 [+0.33↑]	81.27	82.33 [+1.06↑]	81.20	81.47 [+0.27↑]
Validation 2	acc [%]	79.40	79.52 [+0.12↑]	81.96	82.20 [+0.24↑]	81.83	83.33 [+1.50↑]
Test 1	acc [%]	79.50	80.34 [+0.84↑]	79.92	81.59 [+1.67↑]	79.50	81.59 [+2.09↑]
	F1 [%]	71.03	76.19 [+5.16↑]	76.96	77.78 [+0.82↑]	76.82	78.71 [+1.89↑]
Test 2	acc [%]	61.80	63.95 [+2.15↑]	62.66	65.24 [+2.58↑]	64.38	62.23 [-2.15↓]

Table 13: The influence of the dataset's maximum image edge length (300 pixels and 900 pixels) on the search relevance classification quality.

Model name	Metric	Gemma-3-4B-IT-RLV		Gemma-3-12B-IT-RLV		Gemma-3-27B-IT-RLV	
Explanation generation		w/o	w/	w/o	w/	w/o	w/
Validation 1	acc [%]	78.20	79.33 [+1.13↑]	78.67	82.33 [+3.66↑]	80.60	81.47 [+0.87↑]
Validation 2	acc [%]	77.45	79.52 [+2.07↑]	80.45	82.20 [+1.75↑]	80.82	83.33 [+2.51↑]
Test 1	acc [%]	79.08	80.34 [+1.26↑]	80.33	81.59 [+1.26↑]	80.33	81.59 [+1.26↑]
	F1 [%]	68.32	76.19 [+7.87↑]	72.62	77.78 [+5.16↑]	70.47	78.71 [+8.24↑]
Test 2	acc [%]	60.09	63.95 [+3.86↑]	65.24	65.24 [-]	63.95	62.23 [-1.72↓]

Table 14: The influence of finetuning the models on the examples without and with explanations on the search relevance classification quality. Models trained without explanations are trained only on the relevance scores.

Model name	Metric	Gemma-3-4B-IT-RLV		Gemma-3-12B-IT-RLV		Gemma-3-27B-IT-RLV	
Image usage		w/o	w/	w/o	w/	w/o	w/
Validation 1	acc [%]	71.80	78.20 [+6.4↑]	77.60	78.67 [+1.07↑]	79.13	80.60 [+1.47↑]
Validation 2	acc [%]	74.17	77.45 [+3.28↑]	78.26	80.45 [+2.19↑]	79.76	80.82 [+1.06↑]
Test 1	acc [%]	72.38	79.08 [+6.7↑]	78.66	80.33 [+1.67↑]	77.41	80.33 [+2.92↑]
	F1 [%]	62.24	68.32 [+6.08↑]	65.33	72.62 [+7.29↑]	70.09	70.47 [+0.38↑]
Test 2	acc [%]	61.37	60.09 [-1.28↓]	63.09	65.24 [+2.15↑]	61.80	63.95 [+2.15↑]

Table 15: The influence of finetuning the models without and with item images on the search relevance classification quality. Since the explanation contains information from the item images, both models are trained without explanations.

greater improvement is achieved for accuracy on both Validation 2 and Test 1 datasets. Also, the results for 12B and 27B models remain similar, while those for the 4B model remain weaker. Higher resolution seems especially helpful for more difficult predictions. The especially strong F1 gain for the 4B model on Test 1 and accuracy metric on Test 2 for both 4B and 12B models suggests that larger images may improve recognition of subtle visual patterns and special cases.

G.3 Explanation Generation

A comparison of the results obtained on the evaluation data for the created models finetuned on the datasets without and with explanations is presented in Table 14. The benefits of training the models

on examples with explanations are particularly pronounced on the edge cases from Test 1. This is plausible when training on explanations enables the model to better capture subtle differences in the input data. More broadly, finetuning with explanations improves performance for most of the models and dataset variants.

G.4 Image usage

A comparison of the results obtained for the models finetuned without and with item images is presented in Table 15. To ensure a fair comparison that isolates a single missing input signal, we train both models without explanations. This design choice is important because, in most cases, the explanation itself contains information derived from the item images. Excluding explanations from training prevents the model from learning to hallucinate visual information that is unavailable at inference time.

This study shows that images are an important modality that usually improves performance. The benefit is most pronounced for the smallest model, with 4B parameters, which achieves the largest performance gains from incorporating images across all datasets, except Test 2, relative to the 12B and 27B models.

H Cost comparison

A comparison of the cost per 1 million input and 1 million output tokens is provided in Table 16. Token cost for `GPT-4.1` is obtained from the OpenAI pricing page [OpenAI 2026]. Assuming 120 input text tokens, 256 input image tokens, and 130 output text tokens per request in the explanation-generation setup, and using the measured `Gemma-3-12B-IT-RLV-FP8` throughput on a single H100 as a proxy, the estimated pure compute cost without any prompt caching is about $11.86—$19.80 per 1M requests when using an **external** GPU provider [Lambda 2026; Runpod 2026]. Per 1 million tokens, this corresponds to about $0.0234-$0.0391 for both 1 million input and output tokens, assuming maximum GPU utilization. In the label-only setup without explanation generation, the estimated cost is about $2.45—$4.08 per 1M requests, and $0.0065-$0.0108 for both 1 million input and 1 million output tokens. When using dedicated in-house GPUs, these savings are naturally even more substantial.

Model name	`GPT-4.1`	`Gemma-3-12B-IT-RLV-FP8`	
Setup	—	w/o explan.	w/ explan.
1M in. tokens	$2	$0.0065-$0.0108	$0.0234-$0.0391
1M out. tokens	$8		

Table 16: Comparison of cost per 1 million input and 1 million output tokens for `GPT-4.1` and hosted `Gemma-3-12B-IT-RLV-FP8` on an external GPU provider in the label-only- and explanation-generation setups.