QUPID: Quantified Understanding for Enhanced Performance, Insights, and Decisions in Korean Search Engines

Ohjoon Kwon*, **Changsu Lee***, **Jihye Back**, **Lim Sun Suk**,
Inho Kang, **Donghyeon Jeon**†
Naver Corporation
{ohjoon.kwon, changsu.lee, 1oojihye, dongle.75,
once.ihkang, donghyeon.jeon}@navercorp.com

Abstract

Large language models (LLMs) have been widely used for relevance assessment in information retrieval. However, our study demonstrates that combining two distinct small language models (SLMs) with different architectures can outperform LLMs in this task. Our approach—QUPID—integrates a generative SLM with an embedding-based SLM, achieving more accurate relevance judgments while reducing computational costs compared to state-of-the-art LLM solutions. This computational efficiency makes QUPID highly scalable for real-world search systems processing hundreds of millions of queries daily. In experiments across diverse document types, our method demonstrated consistent performance improvements (Cohen's Kappa of 0.646 versus 0.387 for leading LLMs) while offering up to 60x faster inference. Furthermore, when integrated into production search pipelines, QUPID improved nDCG@5 scores by 1.9%. These findings underscore how architectural diversity in model combinations can significantly enhance both search relevance and operational efficiency in information retrieval systems.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in Information Retrieval (IR) tasks, including query-document relevance assessment (Li et al., 2024b; Zhu et al., 2024; Tang et al., 2024). Their strong contextual understanding allows them to approximate human-level judgments in ranking search results and evaluating query modifications. However, deploying LLMs in production environments comes with significant challenges. Latency issues pose the most critical barrier for large-scale search systems, where real-time responses are essential, followed by the high computational cost and substantial memory footprint, making them impractical for environments operating on a scale of hundreds of millions of queries daily(Strubell et al., 2019; Sharir et al., 2020; Howell et al., 2023; Thomas et al., 2024). Furthermore, maintaining and continuously updating such models is resource-intensive, limiting their feasibility in dynamic and multilingual search environments.

To address these challenges, Small Language Models (SLMs) have emerged as a promising alternative. SLMs offer a more cost-effective solution while maintaining competitive performance in various NLP tasks (Xie et al., 2024; Brei et al., 2024; Xu et al., 2025). However, existing approaches to SLM-based relevance assessment face two critical limitations: (1) Single SLM approaches lack the expressive power and contextual understanding of LLMs, resulting in suboptimal relevance judgments in complex queries; and (2) Current ensemble methods primarily utilize multiple instances of the same model architecture (Rahmani et al., 2024a), failing to leverage the complementary strengths that different model architectures could provide.

In this study, we propose QUPID (Quantified Understanding for Enhanced Performance, Insights, and Decisions), a novel relevance assessment approach that directly addresses these limitations by combining two architecturally distinct SLMs in a heterogeneous ensemble. Unlike prior ensemble methods that aggregate multiple instances of the same model, our approach integrates a generative SLM (QUPID_{GEN}), which excels at contextual reasoning through token probabilities, with an embedding-based SLM (QUPID_{EMB}), which captures semantic similarity through dense vector representations. By leveraging both generative reasoning and embedding-based similarity within a unified framework, QUPID outperforms state-of-the-art LLMs and SLM ensemble baselines while significantly reducing computational cost. Our results

*Equal contribution
†Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 541–552
July 28-30, 2025

show that this heterogeneous ensemble strategy enhances relevance labeling accuracy, thus making it a scalable and efficient solution for real-world search engines.

Beyond accuracy improvements, our model has broad practical implications in modern search systems. We demonstrate how QUPID can be integrated into various workflows, including filtering low-quality query-document pairs, evaluating query rewriting and completion modules, assessing the quality of search result snippets, and enhancing ranking models.

To summarize, our contributions are as follows:

- We introduce QUPID, the first relevance labeling approach leveraging a heterogeneous ensemble of generative and embedding-based SLMs, highlighting the effectiveness of architectural diversity.
- We demonstrate that this approach outperforms state-of-the-art LLM-based methods by up to 67% in terms of Cohen's Kappa (κ = 0.646 vs. κ = 0.387), while offering 60x faster inference times (62ms vs. 3258ms), making it highly scalable for production environments.
- We present concrete use cases showcasing how QUPID can be seamlessly integrated into search engine pipelines, yielding measurable improvements in retrieval quality metrics (e.g., +1.9% in nDCG@5) and enhancing overall user satisfaction in real-world deployments.

2 Related Works

LLM-Based Relevance Labeling Recent studies have explored the potential of Large Language Models (LLMs) in relevance assessment tasks, leveraging their strong contextual reasoning and generation capabilities (Li et al., 2024b; Thomas et al., 2024; Upadhyay et al., 2024). For example, Thomas et al. (2024) demonstrated that LLMs could achieve near-human performance in evaluating query-document relevance. However, their proprietary, in-house model setup makes exact replication difficult, limiting reproducibility in real-world applications. Similarly, Upadhyay et al. (2024) introduced an open-source toolkit for evaluating LLM-based relevance labeling models, but their results revealed significant computational inefficiencies, especially when applied to high-throughput search systems. Moreover, these LLM-based methods generally perform well in English but show suboptimal results in multilingual environments such as Korean, highlighting the need for lighter, language-specific models for more efficient deployment (Robinson et al., 2023; Bang et al., 2023; Nguyen et al., 2024; Li et al., 2024c; Jayakody and Dias, 2024).

Embedding-Based Relevance Labeling in LLMs Beyond text generation, recent work has shown that decoder-only LLMs can also be used for embedding-based retrieval tasks (Ma et al., 2023; Wang et al., 2024; Li et al., 2024a). These models, originally trained for next-token prediction, exhibit surprising representation learning capabilities that can be leveraged for similarity-based tasks. However, most existing studies in this domain have focused on fine-tuning embedding models for generalized text similarity, rather than specifically optimizing for query-document relevance labeling.

To address these limitations, several approaches have attempted to enhance LLM embeddings by modifying the attention mechanisms or introducing hard-negative mining techniques (Wang et al., 2024; Li et al., 2024a). Building on this line of work, we integrate embedding capabilities into our SLM ensemble, making it the first hybrid model combining generative and embedding-based SLMs for relevance assessment. This allows our approach to leverage both explicit generation-based judgments and implicit similarity-based signals, improving robustness and accuracy. By uniting explicit generation-based judgments (which capture contextual cues and semantic coherence) with implicit similarity-based signals (which highlight distributional proximity between query and document), our model can more accurately reflect both high-level language understanding and fine-grained relational cues.

SLM-Based Ensembles for Relevance Assessment To address the high computational costs of LLMs, recent studies have explored ensemble methods using Small Language Models (SLMs). For example, JudgeBlender (Rahmani et al., 2024a) aggregates multiple generative SLMs to improve relevance labeling accuracy. However, this approach requires long prompts and a two-step inference process, leading to higher inference latency (1000ms+) and resource consumption. Furthermore, it focuses solely on generation-based relevance assessment without incorporating embedding-based similarity, potentially limiting its effectiveness in ranking tasks. This synergy yields

richer feature representations and mitigates the limitations of using either approach alone, ultimately enhancing both robustness and accuracy in query-document relevance tasks.

Unlike these prior works, our approach introduces a heterogeneous ensemble of two distinct SLM architectures, where one model specializes in generative reasoning and the other in embedding-based relevance computation. By combining both generative and representation-learning capabilities, our method achieves higher accuracy with significantly reduced computational cost, making it scalable for real-world search systems.

3 Methodology

In this section, we introduce our approach to relevance labeling for query-document pairs using two small-scale language models (SLMs). We first describe the process of creating and cleaning our dataset (Section 3.1). We then employ one SLM as a generative relevance labeling model and another SLM as an embedding-based relevance labeling model (detailed in Sections 3.2 and 3.3, respectively). Finally, we describe our ensemble strategy that combines both models' outputs to achieve higher accuracy and robustness (Section 3.4).

3.1 Data Curation

Our approach to curating training data for relevance labeling consists of two main strategies: collecting real-world document data and generating synthetic hard-negative samples. By integrating these approaches, we ensure that our dataset is diverse, well-balanced, and robust to generalization challenges.

3.1.1 Real-World Document Collection

We collected various query-document pairs from the web. These pairs come from three main sources to capture a broad range of text lengths, styles, and complexities:

- Snippet-Style Web Content: Short pieces of text—often lists, tables, or brief paragraphs—extracted from search engine snippets.
- User-Generated Content: Texts directly provided or created by users, such as forum posts or community Q&A entries.
- General Web Documents: Freely crawled online documents covering diverse topics and domains.

With more than 850K query-text pairs, experienced human annotators[1] carefully review each pair and assign an appropriate label among four classes (see Appendix A.1 for details). We adopt a voting scheme to resolve any disagreements, and if all three annotators assign different labels, an additional linguist overseeing the annotation process makes the final judgment. By including snippet-style, user-generated, and general web documents, we ensure varied document lengths and structures such as text, lists, and tables, which is vital for robust model training. (Note that *Somewhat Relevant* has been classified as a relevant label.)

3.1.2 Synthetic Hard-Negatives Generation

Existing studies in retrieval and embedding emphasize the importance of hard-negative samples to enhance model robustness and generalization (Lee et al., 2024; de Souza P. Moreira et al., 2024; Aho and Ullman, 1972; Wang et al., 2024). We follow a similar strategy to further refine our training data.

For each query-document pair collected in Section 3.1.1, we prompt the model to generate new documents that are superficially similar to the original but contextually off-topic or partially misleading. We use Mistral2-Large to produce additional hard-negative documents because it shows plausible Korean synthetic data based on our internal evaluation. Further details on the prompt design can be found in Appendix A.2.

By combining real-world data (approximately 48.70% web documents, 12.17% user-generated content, and 24.62% snippets) with synthetically generated hard negatives (14.51%), we construct a curated dataset comprising roughly 1M query-document pairs.

3.2 Generative Relevance Labeling Model

SLM as a Relevance Labeler When using a generative model for relevance labeling in a query-document setting, there are generally two approaches. One is to directly interpret the tokens produced by the large language model (LLM) as the final decision, optionally appending a confidence estimation step (Thomas et al., 2024; Ni et al., 2025). Another approach is to leverage the token probabilities associated with a specific label or query (Sachan et al., 2023; Zhuang et al.,

[1]The annotators are part of a specialized data-labeling company and work under close guidance and feedback from linguists with domain expertise.

2024), thereby introducing a more explicit calibration stage.

In this work, we follow the token-probability approach proposed by Zhuang et al. (2024), rather than interpreting directly sampled tokens. Specifically, for each query-document pair (q_i, d_i), our generative model M_{gen} is fine-tuned to produce exactly one among three special tokens $\{l_1, l_2, l_3\}$. Each token corresponds to a distinct relevance category (e.g., Highly Relevant, Low Relevance, Not Relevant). We then obtain the log probability of each label token and convert these values into probabilities via a softmax function as follows:

$$p_{i,k} = \mathrm{M}_{\text{gen}}(l_k \mid q_i, d_i), \tag{1}$$

be the probability that the model assigns to label l_k. We define the final relevance score as:

$$f(q_i, d_i) = \sum_{k=1}^{K} p_{i,k} \cdot y_k, \tag{2}$$

where y_k is a label-specific weight (or numeric value) that can be tuned based on downstream requirements or validation set performance. In our experiments, we set these values as follows: *relevant* $= 1.0$, *somewhat relevant* $= 0.5$, and *irrelevant* $= 0$. This approach provides a more informative representation of the model's confidence by considering the probability distribution over possible labels rather than relying on a single sampled output. This allows for a more nuanced understanding of the model's uncertainty, which can be particularly useful in downstream applications requiring risk-aware decision-making.

3.3 Embedding-based Relevance Labeling Model

For fine-tuning the embedding-based model, M_{emb}, it takes a query-document pair (q, d) as input and generates a relevance score through an embedding extraction and classification process. Given an input pair, the model produces a sequence of token-level hidden state embeddings:

$$H = M_{\text{emb}}(q, d) = [h_1, h_2, \ldots, h_n], \tag{3}$$

where $h_i \in \mathbb{R}^{d_h}$ represents the hidden state of the i-th token, and d_h denotes the dimensionality of the model's hidden representation. To obtain a fixed-size representation, we apply mean pooling as it showed more stable performance in our experiments and prior research (Lee et al., 2025; de Souza P. Moreira et al., 2025):

$$\mathbf{h}_{\text{agg}} = \frac{1}{n} \sum_{i=1}^{n} h_i. \tag{4}$$

A linear transformation is then applied to map $\mathbf{h}_{\text{agg}}$ into a relevance score vector:

$$s = W \mathbf{h}_{\text{agg}} + b, \tag{5}$$

where $W \in \mathbb{R}^{K \times d_h}$ is the learned weight matrix, $b \in \mathbb{R}^K$ is the bias term, and K represents the number of predefined relevance categories. Finally, the softmax function is applied to compute the probability distribution over the relevance labels.

3.4 Score Ensemble

To enhance the robustness of relevance scoring, we combine the outputs of the generative model M_{gen} and the embedding-based model M_{emb} through a weighted averaging approach. Weighted averaging preserves the independence of each model's outputs and provides a straightforward way to balance generative vs. embedding signals Each model produces a relevance score given a query-document pair (q, d):

$$s_{\text{gen}} = M_{\text{gen}}(q, d), \tag{6}$$

$$s_{\text{emb}} = M_{\text{emb}}(q, d). \tag{7}$$

To obtain the final ensemble relevance score, we compute a weighted sum of these two scores:

$$s_{\text{final}} = w_{\text{gen}} \cdot s_{\text{gen}} + w_{\text{emb}} \cdot s_{\text{emb}}, \tag{8}$$

where w_{gen} and w_{emb} are the weighting coefficients assigned to the generative and embedding-based models, respectively. The weights can be tuned based on validation performance or set heuristically. In practice, we determine the optimal w_{gen} and w_{emb} by evaluating the ensemble's effectiveness on a held-out validation set, selecting the combination that maximizes relevance prediction accuracy.

4 Experimental Setup

In this section, we present a comprehensive evaluation environment of our model in diverse experimental settings. Our model is fine-tuned on HCX-S (Yoo et al., 2024), a state-of-the-art instruct-tuned model known for its superior performance in Korean-language tasks.

4.1 Dataset

To evaluate robustness across document types, we build test sets comprising snippet-style web content (3,000 samples; 2,268 relevant, 732 irrelevant), user-generated content (20,000 samples; 11,148 relevant, 8,852 irrelevant), and general web documents (9,000 samples; 4,971 relevant, 4,029 irrelevant).

4.2 Baselines

To evaluate the effectiveness of our proposed hybrid ensemble approach, we compare our models against two categories of baselines: (1) representative large language models (LLMs) and (2) SLM ensemble models.

4.2.1 Representative LLMs

We evaluate a set of representative LLMs (LLaMA, Mistral, and Qwen), each capable of performing relevance labeling via zero-shot inference. Various prompting strategies exist for instructing LLMs in relevance assessment (Sun et al., 2023; Faggioli et al., 2023; Farzi and Dietz, 2024; Thomas et al., 2024), and we experiment with the representative prompting method showed best performance in LLMJudge benchmark (Rahmani et al., 2024b). For more information, see the Table 3 in Farzi and Dietz (2024)

4.2.2 SLM Ensemble Method

We also compare our approach to JudgeBlender (Rahmani et al., 2024a). It follows a multi-model ensembling strategy where each constituent model is prompted separately for relevance assessment, and their outputs are combined to improve robustness. The same prompts used in the representative LLMs experiments were employed. Based on the observation that LLaMA models performs poorly in Korean, we replaced it with the HCX-S (Yoo et al., 2024).

5 Results

5.1 Quantitative Results

Table 1 reveals that representative large language models (LLMs) face challenges in capturing the precise relevance between queries and documents. LLaMA3.3-70b shows notably lower performance, likely due to its weaker multilingual capabilities, especially in Korean. To provide a comprehensive evaluation of model effectiveness, we employed AUC to assess how well each model ranks relevant documents against irrelevant ones across varying thresholds. Additionally, we used Cohen's kappa to measure the agreement level between model predictions and human-annotated relevance labels, offering insights into how closely automated labeling aligns with human judgment.

While the ensemble methods with SLMs show some improvement, they are still not enough to replace human judgment or reliably assess search quality. The two-step inference and ensemble approach based on three SLMs of approximately 8B parameters fell short of the zero-shot prompting performance of the LLMs. This suggests that a simple SLM ensemble, without target-specific fine-tuning or heterogeneous model combinations, cannot match the capacity of LLMs.

Our proposed model consistently outperforms these leading LLMs and SLM blending methods. Moreover, Table 2 demonstrates that the combination of heterogeneous models in an ensemble leads to substantial performance improvements, highlighting the advantage of leveraging diverse model architectures to enhance task effectiveness.

5.2 Use Cases and Efficiency

We examine several practical use cases demonstrating how our QUPID can be seamlessly integrated into real-world search engine workflows. These particular use cases—(1) filtering low-quality pairs, (2) evaluating query rewriting and completion, (3) assessing snippet quality, and (4) improving ranking—were selected because they represent common challenges in large-scale search systems and highlight different facets of relevance assessment.

Filtering low-quality Q-D pairs Search engines pre-assign documents to frequently occurring or time-sensitive queries to enable faster response times (Nogueira et al., 2019; Wen et al., 2023). Since these documents often appear at the top of search results with high confidence, ensuring their relevance and quality is crucial. As shown in Appendix A.4.1 and Figure 2, QUPID plays a vital role in identifying and filtering out low-quality content before it reaches users, achieving a precision of over 0.9.

Evaluating Query Refinement Modules Our relevance model provides an automatic approach to evaluating query rewriting (Wu et al., 2022; Sun et al., 2024; Liu and Mozafari, 2024) or completion models (Jaech and Ostendorf, 2018; Kim, 2019; Gog et al., 2020). Effectiveness of these models

Model	Cohen's Kappa (κ)				AUC (Relevant / Irrelevant)			
	UGC	**Snippet**	**Web-D**	Avg.	**UGC**	**Snippet**	**Web-D**	Avg.
Representative LLMs								
ChatGPT-4o	0.224	0.424	0.514	0.387	0.696 / 0.622	0.915 / 0.511	0.818 / 0.760	0.810 / 0.631
LLaMA3.3-70b-instruct	0.129	0.256	0.402	0.262	0.667 / 0.141	0.883 / 0.229	0.762 / 0.533	0.771 / 0.301
Mistral-large-instruct-2411	0.154	0.326	0.458	0.313	0.713 / 0.632	0.924 / 0.578	0.812 / 0.751	0.816 / 0.654
Qwen-2.5-72b-instruct	0.160	0.333	0.436	0.310	0.721 / 0.654	0.954 / 0.574	0.821 / 0.762	0.832 / 0.663
SLM Ensemble Model								
Mistral-8b-instruct-2410	0.151	0.119	0.257	0.176	0.695 / 0.542	0.781 / 0.363	0.634 / 0.494	0.703 /0.466
Qwen-2.5-7b-instruct	0.124	0.262	0.318	0.235	0.698 / 0.379	0.839 / 0.419	0.711 / 0.564	0.749 / 0.454
HCX-S	0.157	0.174	0.298	0.210	0.692 / 0.547	0.783 / 0.415	0.645 / 0.564	0.707 / 0.509
JudgeBlender								
+MV(Avg.)	0.207	0.269	0.291	0.256	0.696 / 0.640	0.867 / 0.472	0.686 / 0.710	0.750 / 0.607
+MV(Rand.)	0.162	0.183	0.298	0.214	0.633 / 0.482	0.783 / 0.396	0.595 / 0.633	0.670 / 0.504
+AV	0.154	0.278	0.331	0.254	0.652 / 0.623	0.881 / 0.494	0.689 / 0.724	0.741 / 0.614
Ours (fine-tuned)								
QUPID$_{GEN}$	0.582	0.418	0.674	0.558	0.897 / 0.880	0.932 / 0.707	0.960 / 0.940	0.930 / 0.842
QUPID$_{EMB}$	0.662	0.518	0.590	0.590	0.927 / 0.892	0.904 / 0.715	0.889 / 0.851	0.907 / 0.819
QUPID$_{ENSEMBLE}$	**0.679**	**0.569**	**0.783**	**0.646**	**0.929 / 0.911**	**0.944 / 0.756**	**0.962 / 0.946**	**0.945 / 0.871**

Table 1: Evaluation results across models on three document types. Cohen's Kappa (κ) and AUC scores are reported. AUC is reported as Relevant AUC / Irrelevant AUC.

Model	Avg. κ	Avg. AUC
QUPID$_{GEN}$	0.558	0.930 / 0.842
QUPID$_{GEN*3}$	0.564	0.932 / 0.849
QUPID$_{GEN*5}$	0.569	0.933 / 0.851
QUPID$_{EMB}$	0.590	0.907 / 0.819
QUPID$_{EMB*3}$	0.597	0.913 / 0.832
QUPID$_{EMB*5}$	0.607	0.913 / 0.835
QUPID$_{ENSEMBLE}$	**0.646**	**0.945 / 0.871**

Table 2: The rows above show the results of ensembling with the same architecture that were trained using the same approach but with different hyperparameters.

should be assessed based on whether they improve the search results. As illustrated in Table 4, we utilize our relevance model to evaluate the performance of the LLM-based query generation modules operating in our search engine.

Evaluating the quality of snippets extracted from documents Our relevance model can independently evaluate query-document (Q-D) relevance and query-snippet (Q-S) relevance. If a document has a high relevance score but a low snippet relevance score, it suggests that the document itself is relevant to the query, but the extracted snippet is misleading or unrepresentative. As illustrated in Table 5, such cases often lead to inaccurate search summaries, which can negatively impact the user experience.

Applying QUPID for Search Results Ranking To further evaluate the benefits of QUPID in a real-world ranking scenario, we tested the model on the ranking task. Table 7 shows the ranking metrics achieved by the baseline ranking model and our proposed method. Having explored the potential of replacing the existing ranking model, we actively utilize QUPID as the ranking model in our search engine.

Efficiency Compare For efficiency evaluation, we measured the latency of each model. Each model was deployed and served on the same A100-80G GPUs with vLLM serving engine. Due to a significantly shorter system prompt and generating at most a single token, the QUPID model exhibits substantially lower latency.

Model	Sys. Prompt	Latency
QUPID$_{ENSEMBLE}$	10 token	62 ms
JudgeBlender	362 tokens	1173 ms
LLaMA3.3-70b	353 tokens	3258 ms
Qwen-2.5-72b	384 tokens	3520 ms
Mistral-large	392 tokens	3690 ms

Table 3: The input tokens for JudgeBlender are the average of the three models. When results from multiple models are required, they are obtained asynchronously through parallel calls. Refer to Appendix A.4.2.

Trump -> *Trump assassination attempt*		**Biden -> *Biden assassination attempt***	
Original Query	Trump	**Original Query**	Biden
Auto-Completion	Trump assassination attempt	**Auto-Completion**	Biden assassination attempt
Relevance Score	Rank 1/2/3: 0.804/0.905/0.384	**Relevance Score**	Rank 1/2/3: 0.307/0.219/0.135
Top-3 Retrieved Documents (Title & Body)			
Rank 1 (Title)	"Donald Trump rally attack incident"	**Rank 1 (Title)**	"Trump: 'Assassination attempt due to Biden-Harris rhetoric'"
Rank 1 (Body)	"On July 13, 2024, in Pennsylvania, an assassination attempt was made on Donald Trump during a campaign rally. Security forces responded immediately and neutralized the attacker."	**Rank 1 (Body)**	"On September 17, 2024, in a Fox News interview, Trump claimed that Biden and Harris were responsible for inciting violence, linking their rhetoric to the assassination attempt."
Rank 2 (Title)	"Trump, second assassination attempt... Secret Service responded in time"	**Rank 2 (Title)**	"Breaking: Trump, second assassination attempt suspect arrested at golf course"
Rank 2 (Body)	"On September 16, 2024, Secret Service intervened in Florida to prevent another assassination attempt on Trump. The suspect was found carrying a weapon near his residence."	**Rank 2 (Body)**	"Breaking reports suggest a suspect was arrested at a golf course while attempting another attack on Trump. Officials confirmed White House was briefed immediately."
Rank 3 (Title)	"'Another assassination threat due to Harris' – Trump's claim had no impact on election dynamics"	**Rank 3 (Title)**	"'Pro-Trump' Musk mocks Biden and Harris over assassination rumors"
Rank 3 (Body)	"Trump suggested that Harris' political influence contributed to threats against him, though polls indicated minimal impact on voter sentiment."	**Rank 3 (Body)**	"Elon Musk commented on X (formerly Twitter) that no one had ever attempted to assassinate Biden or Harris, sparking controversy online."

Table 4: Comparison of Top-3 Retrieved Documents (Title & Body) for Trump and Biden Query Auto-Completion Cases. It can be observed that the relevance score is significantly higher when the auto-completed query corresponds to a realistically searchable query (left) compared to when it does not (right). The text shown in the table was directly used as input, and although the majority of the training data is in Korean, the multilingual capability of the backbone model appears to enable its functionality in English as well.

Query: **What companies does Elon Musk own?**	
Document **(Q-D score: 0.812)**	Extracted Snippet **(Q-S score: 0.284)**
Elon Musk's Business Empire: A Look at His Companies. Elon Musk is one of the most influential entrepreneurs of the 21st century... **His ventures range from electric vehicles and renewable energy to space exploration and artificial intelligence. He serves as CEO of Tesla, SpaceX, and Neuralink**...	Musk was born in South Africa and later moved to the U.S. to pursue his career. From an early age, he showed a strong interest in technology and entrepreneurship.

Table 5: Side-by-side comparison of a document and its extracted snippet. The document is relevant to the query (high Q-D relevance score), but the snippet is misleading (low Q-S relevance score). The relevant information to the query is highlighted in bold. We note that the relevance scores shown in this table were obtained by directly feeding the English text as input to the QUPID model.

6 Conclusion

In this paper, we introduced QUPID, a heterogeneous ensemble of generative and embedding-based SLMs for relevance labeling. Our approach improves accuracy while reducing computational costs and demonstrates practical applicability in real-world search engines. These findings highlight the value of architectural diversity in enhancing relevance assessment while maintaining efficiency.

7 Limitations

While our proposed QUPID approach demonstrates significant improvements in relevance labeling efficiency and accuracy, it has several limitations that warrant further investigation.

Text Modality Only Our current approach is designed exclusively for text-based relevance assessment and does not incorporate multimodal capabilities. Many modern search applications involve a combination of text, images, and videos, where relevance cannot always be determined through textual information alone. The lack of image and video processing limits the applicability of our method in broader search engine contexts, particularly in domains such as e-commerce, news aggregation, and multimedia search.

Limited Feature Scope Our model primarily focuses on relevance as the key criterion for assessing search results. However, in real-world applications, additional factors such as readability, aesthetic appeal, and trustworthiness also play a critical role in determining the overall quality of retrieved documents. For instance, a document may be highly relevant but poorly formatted or difficult to read, negatively impacting user experience. Future iterations of our model should incorporate auxiliary scoring mechanisms to address these qualitative aspects of search quality.

Despite these limitations, our findings establish a strong foundation for efficient and scalable relevance assessment. Addressing these challenges in future research—particularly through multimodal expansion and the inclusion of richer feature representations—will further enhance the practicality and robustness of our approach.

Acknowledgments

We would like to express our heartfelt gratitude to the labeling experts for their invaluable assistance in reviewing and verifying the solutions presented in this work. Their detailed feedback, as well as the efforts of all other contributors, significantly improved the quality and accuracy of our results. We confirm that each contributor was duly compensated for their time and expertise. Additionally, we utilized generative AI tools to assist with code development and to refine the manuscript's English expression. The authors assume full responsibility for the content herein, including any errors that may remain.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. The Theory of Parsing, Translation and Compiling, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. Preprint, arXiv:2302.04023.

Felix Brei, Johannes Frey, and Lars-Peter Meyer. 2024. Leveraging small language models for text2sparql tasks to improve the resilience of ai assistance. Preprint, arXiv:2405.17076.

Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. Nv-retriever: Improving text embedding models with effective hard-negative mining. Preprint, arXiv:2407.15831.

Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2025. Nv-retriever: Improving text embedding models with effective hard-negative mining. Preprint, arXiv:2407.15831.

Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast et al. 2023. Perspectives on large language models for relevance judgment. In Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '23, page 39–50. ACM.

Naghmeh Farzi and Laura Dietz. 2024. Best in tau@llmjudge: Criteria-based relevance evaluation with llama3. Preprint, arXiv:2410.14044.

Simon Gog, Giulio Ermanno Pibiri, and Rossano Venturini. 2020. Efficient and effective query auto-completion. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, page 2271–2280. ACM.

Kristen Howell, Gwen Christian, Pavel Fomitchov, Gitit Kehat, Julianne Marzulla, Leanne Rolston, Jadin Tredup, Ilana Zimmerman, Ethan Selfridge, and Joseph Bradley. 2023. The economic trade-offs of large language models: A case study. Preprint, arXiv:2306.07402.

Aaron Jaech and Mari Ostendorf. 2018. Personalized language model for query auto-completion. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 700–705, Melbourne, Australia. Association for Computational Linguistics.

Ravindu Jayakody and Gihan Dias. 2024. Performance of recent large language models for a low-resourced language. Preprint, arXiv:2407.21330.

Gyuwan Kim. 2019. Subword language model for query auto-completion. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5022–5032, Hong Kong, China. Association for Computational Linguistics.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nv-embed: Improved techniques for training llms as generalist embedding models. Preprint, arXiv:2405.17428.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia et al. 2024. Gecko: Versatile text embeddings distilled from large language models. Preprint, arXiv:2403.20327.

Chaofan Li, Zheng Liu, Shitao Xiao, Yingxia Shao, and Defu Lian. 2024a. Llama2Vec: Unsupervised adaptation of large language models for dense retrieval. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3490–3500, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2024b. From matching to generation: A survey on generative information retrieval. Preprint, arXiv:2404.14851.

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2024c. Language ranker: A metric for quantifying llm performance across high and low-resource languages. Preprint, arXiv:2404.11553.

Jie Liu and Barzan Mozafari. 2024. Query rewriting via large language models. Preprint, arXiv:2403.09060.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. Preprint, arXiv:2310.08319.

Xuan-Phi Nguyen, Sharifah Mahani Aljunied, Shafiq Joty, and Lidong Bing. 2024. Democratizing llms for low-resource languages by leveraging their english dominant abilities with linguistically-diverse prompts. Preprint, arXiv:2306.11372.

Jingwei Ni, Tobias Schimanski, Meihong Lin, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2025. Diras: Efficient llm annotation of document relevance in retrieval augmented generation. Preprint, arXiv:2406.14162.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. Preprint, arXiv:1904.08375.

Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, and Bhaskar Mitra. 2024a. Judgeblender: Ensembling judgments for automatic relevance assessment. Preprint, arXiv:2412.13268.

Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles L. A. Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. 2024b. Llmjudge: Llms for relevance judgments. Preprint, arXiv:2408.08896.

Nathaniel Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. ChatGPT MT: Competitive for high- (but not low-) resource languages. In Proceedings of the Eighth Conference on Machine Translation, pages 392–418, Singapore. Association for Computational Linguistics.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2023. Improving passage retrieval with zero-shot question generation. Preprint, arXiv:2204.07496.

Or Sharir, Barak Peleg, and Yoav Shoham. 2020. The cost of training nlp models: A concise overview. Preprint, arXiv:2004.08900.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. Preprint, arXiv:1906.02243.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 14918–14937, Singapore. Association for Computational Linguistics.

Zhaoyan Sun, Xuanhe Zhou, and Guoliang Li. 2024. R-bot: An llm-based query rewrite system. Preprint, arXiv:2412.01661.

Qiaoyu Tang, Jiawei Chen, Zhuoqun Li, Bowen Yu, Yaojie Lu, Cheng Fu, Haiyang Yu, Hongyu Lin, Fei Huang et al. 2024. Self-retrieval: End-to-end information retrieval with one large language model. Preprint, arXiv:2403.00801.

Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. Preprint, arXiv:2309.10621.

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. Umbrela: Umbrela is the (open-source reproduction of the) bing relevance assessor. Preprint, arXiv:2406.06519.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. Preprint, arXiv:2401.00368.

Xueru Wen, Xiaoyang Chen, Xuanang Chen, Ben He, and Le Sun. 2023. Offline pseudo relevance feedback for efficient and effective single-pass dense retrieval. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, page 2209–2214, New York, NY, USA. Association for Computing Machinery.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. Preprint, arXiv:2112.08558.

Yukang Xie, Chengyu Wang, Junbing Yan, Jiyong Zhou, Feiqi Deng, and Jun Huang. 2024. Making small language models better multi-task learners with mixture-of-task-adapters. In Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '24, page 1094–1097, New York, NY, USA. Association for Computing Machinery.

Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. 2025. Slmrec: Distilling large language models into small for sequential recommendation. Preprint, arXiv:2405.17890.

Kang Min Yoo, Jaegeun Han, Sookyo In, Heewon Jeon, Jisu Jeong, Jaewook Kang, Hyunwook Kim, Kyung-Min Kim, Munhyong Kim et al. 2024. Hyperclova x technical report. Preprint, arXiv:2404.01954.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2024. Large language models for information retrieval: A survey. Preprint, arXiv:2308.07107.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. Preprint, arXiv:2310.14122.

A Appendices

A.1 Guidelines for Evaluating Search Results

A.1.1 Evaluation Scope

The evaluation targets documents (text) corresponding to a query. Both the title and body content are considered holistically. The key criterion is whether the title and body together provide sufficient and relevant information to match the user's intent.

A.1.2 Evaluation Criteria

Documents are classified into four categories:

- **Relevant (R)**: Fully relevant and contains sufficient information.
- **Somewhat Relevant (SR)**: Relevant but lacks sufficient details.
- **Irrelevant (I)**: Either irrelevant or missing key information.
- **Not Evaluable**: Cases where the query is unclear or improperly corrected by a search suggestion, making them unsuitable for reliable annotation. As such, they are excluded from both training and evaluation.

A.1.3 Key Evaluation Considerations

- **Title & Body Alignment**: A document is considered high quality if the title and body together sufficiently answer the query.
- **Insufficient Body Content**: Even if the title is relevant, a document is marked as low quality if the body lacks necessary details.
- **Title-Body Discrepancy**: If the title does not match the query but the body contains sufficient information, the document is still rated based on body content.
- **Hashtag-Based Content**: Hashtags alone can be evaluated if they effectively convey information.
- **Non-Text Queries**: Queries containing only numbers or foreign languages are excluded.

A.1.4 Evaluation Process

1. **Relevance Check**: Determines if the document aligns with the user's intent.
    - If unrelated, it is marked **Irrelevant (I)**.
    - If related, move to the next step.

2. **Information Sufficiency Check**:
    - If the document fully answers the query, it is **Relevant (R)**.
    - If additional searches are needed, it is **Somewhat Relevant (SR)**.
    - If the document lacks necessary details entirely, it is **Irrelevant (I)**.
3. **Freshness Requirement**: If the query demands the latest data (e.g., financial, legal, or event-based queries), outdated responses are marked **Irrelevant (I)**.

A.1.5 Handling of Special Cases

- **Ambiguous Queries**: If a query has multiple meanings, the most commonly searched intent is used for evaluation.
- **Search Correction Errors**: If an automatic query correction leads to a mismatched query-document pair, the result is marked **Not Evaluable**.
- **Table/List Format Documents**: If extracted tables contain errors, missing data, or require verification, they are marked for **Further Review**.

This structured framework ensures objective and consistent evaluation of search results.

A.2 Hard Negative Document Generation

To enhance the robustness of our ranking model, we generate hard negative documents by leveraging a structured prompt. The objective is to create documents that contain query-related keywords but deviate in meaning, ensuring they do not fulfill the user's intent. This method helps the model distinguish between relevant and misleading results.

A.2.1 Structured Prompt for Hard Negative Document Generation

We utilize the following structured prompt to systematically generate hard negative documents. This prompt ensures that the generated documents resemble real-world documents while maintaining low relevance to the given query.

> **System Prompt:** You are an AI search system optimized for retrieving relevant information based on user queries.
>
> **Instruction:** Given a **search query** and its **highest relevance document**, generate a new hard negative document that meets the following criteria:
>
> - The document must contain keywords from the query but use them in a different semantic context.
> - The document should provide useful information, but the information must not align with the query's intent.
> - The document should not contain any direct answers to the query but may include peripheral information.
> - The document must be factually accurate and must not include fabricated or false information.
> - The document should be significantly less relevant to the query compared to the most relevant document.
>
> Additionally, the document's style should match the style of the given relevant document:
>
> - If the most relevant document follows an encyclopedic format, the generated document should also adopt a formal, academic style.
> - If the most relevant document follows a blog-like format, the generated document should adopt a conversational and subjective style.
>
> Ensure that the generated output follows these guidelines and is formatted in JSON with a clear distinction between `title` and `document` fields.

A.2.2 Criteria for Hard Negative Documents

Hard negative documents are generated based on the following criteria:

- **Query Mismatch:** The document discusses a topic that is lexically similar to the query but semantically different.
- **Useful but Irrelevant:** The document contains valuable information but does not directly answer the query.
- **Incomplete Information:** The document provides only partial information, requiring further searches to obtain the full answer.

Hyperparams.	QUPID$_{EMB}$	QUPID$_{GEN}$
LR	1e-5	1e-5
Epochs	5	5
Optimizer	Adam	Adam
GPU (hours)	A100 (440)	A100 (480)
Max Length	1024	1024

Table 6: T: inference temperature. Both model trained on 8xA100-80G.

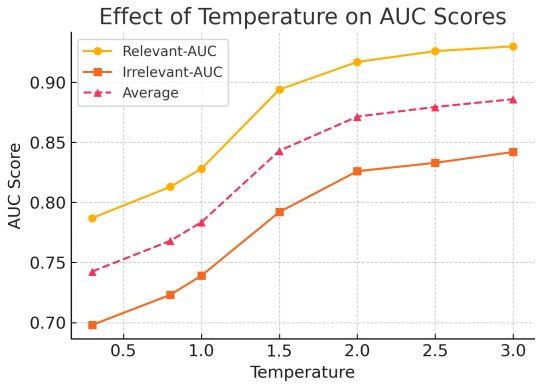

Figure 1: Effect of Temperature on AUC Scores

A.3 Hyper Parameters

In this section, we detail the hyperparameters used for training the embedding-based model (**EMB**) and the generative model (**GEN**) described in our methodology. We also indicate the proportion of synthetic data used, along with any notable implementation remarks. Refer to Table 6.

Remarks. We set the generation temperature to 3.0 during inference. Unlike general-purpose LLM usage where temperatures below 1.5 are typical, we found that raising the temperature up to approximately 3.0 yielded better results in our fine-tuned scenario. We hypothesize that a lower temperature causes the model to be overly confident in a single token (e.g., probability ≈ 0.998) and thus reduces the meaningful effect of aggregating multiple token probabilities. A summary of these experiments and their outcomes is provided in Figure 1. Mean pooling is used for extracting embedding from QUPID$_{EMB}$ (decoder-only model structure).

A.4 Details of Use Case and Efficiency

A.4.1 Filtering low-quality Q-D pairs

Fundamentally, the trade-off between precision and recall is observed in Figure 2. As indicated by the black horizontal dotted line, we can apply thresholding that prioritizes high precision, even at the cost of some coverage (recall). This approach allows us to minimize side effects, such as filtering out high-quality documents, while effectively filtering out low-quality documents with high accuracy (above 0.95).

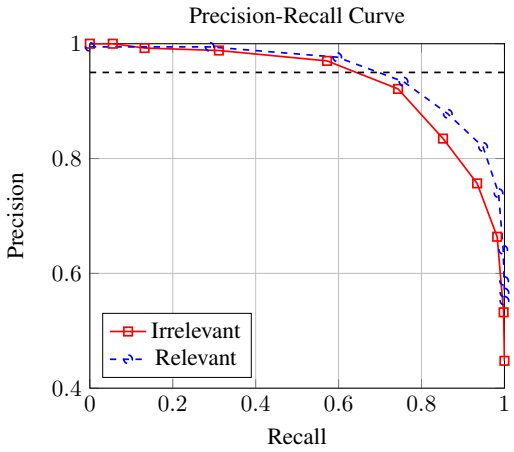

Figure 2: PR curve of QUPID$_{ENSEMBLE}$ model on the **Web-D** dataset.

A.4.2 Efficiency Compare

In contrast to the prompting-based approach, the experiment for our model (QUPID) involved the addition of just one special token (<|*task_prefix*|>) and short system prompt (*query:*, *document:*), alongside the query and document. Through fine-tuning on the target task with a vast amount of data, we experimentally confirmed that lengthy prompts were unnecessary. Thus, through the fine-tuning process, we can also observe the advantage of being able to drastically simplify long natural language prompts.

Metric	Search Results	w/ QUPID
nDCG@1	0.8236	**0.8265**
nDCG@3	0.8511	**0.8651**
nDCG@5	0.8769	**0.8938**
DCG@1	8.7007	**9.0549**
DCG@3	15.6358	**16.1056**
DCG@5	19.3753	**19.9544**

Table 7: Comparison of ranking metrics on 719 queries (each with an average of 15 documents). This demonstrates that the relevance score can directly serve as a ranking feature, even though no separate ranking loss was introduced and the original architecture and training methodology of QUPID were maintained.