Retrieval-Augmented Relevance Judgment for Specialized Domains

Taichi Motegi[1], Makoto P. Kato[1], Kazuhisa Hatakeyama[2] and Yosuke Yurikusa[2,*]

[1]University of Tsukuba, Tsukuba, Japan
[2]MISUMI Group Inc., Tokyo, Japan

Abstract

In this paper, we propose a relevance judgment method called Retrieval-Augmented Relevance Judgment (RAJ), which compensates for the lack of domain knowledge in large language models (LLMs) by utilizing external information through retrieval. In RAJ, documents relevant to the query targeted for relevance judgment are first retrieved. Based on these retrieved documents, a definition sentence for the query is generated using an LLM. This generated definition is then provided to the LLM to perform the relevance judgment in order to improve the accuracy. To evaluate the performance of RAJ, we conducted experiments using the TREC Robust 2004, TREC-COVID, NFCorpus, and MISUMI (an e-commerce site for mechanical parts) datasets. We assessed the judgment accuracy by measuring the level of agreement with human-labeled relevance judgments. The experimental results showed that RAJ achieved higher agreement compared to the condition without retrieval augmentation. Furthermore, RAJ demonstrated greater improvements in domain-specific corpora compared to general corpora, particularly with increased recall for non-relevant documents. The results also suggested that RAJ tends to be more effective when the generated definition is similar to relevant documents.

Keywords

Retrieval–Augmented Generation, Multimodal LLMs, Benchmarking, Evaluation, Information Retrieval

1. Introduction

Relevance judgment is the task of determining whether a document is relevant to the user's information need, and datasets with such judgments are essential for training and evaluating retrieval models. Traditionally, relevance judgments have been conducted manually, which poses a significant cost barrier to constructing large-scale datasets. To address this issue, recent studies have explored the use of large language models (LLMs) for relevance judgment tasks [1, 2]. However, in highly specialized domains, the knowledge that an LLM acquires from pre-training alone may be insufficient for accurate relevance judgment. Relevance judgment grounded in expert knowledge goes beyond ordinary relevance judgment, demanding domain-specific understanding, explicit criteria, and greater rigor in judgment. Prior studies have shown that the accuracy and consistency of LLM outputs tend to decline in fields such as medicine, finance, and law [3, 4, 5]. Although domain-specific approaches for performing relevance judgment with LLMs have been proposed [6, 7, 8], these techniques cannot be directly applied to a wide variety of specialized domains.

To address this challenge, we propose Retrieval-Augmented Relevance Judgment (RAJ), a relevance judgment method that leverages external knowledge retrieved through a search engine. RAJ applies the framework of Retrieval-Augmented Generation (RAG) [9] to the relevance judgment task. RAG supplements LLMs with retrieved external information to generate more informed outputs. Similarly, RAJ retrieves relevant documents based on the input query and then uses the retrieved content to inform the relevance judgment process performed by the LLM. This enables the LLM to make better judgments by compensating for its lack of domain knowledge. As shown in Figure 1, RAJ first retrieves documents

This paper was reviewed and accepted by the program committee for BREV-RAG 2025: Beyond Relevance-based EValuation of RAG Systems, a SIGIR-AP 2025 workshop, held on December 10, 2025 in Xi'an, China.

*Corresponding author.
s2421700@u.tsukuba.ac.jp (T. Motegi); mpkato@acm.org (M. P. Kato); kazuhisa.7z6d.hatakeyama@misumi.co.jp (K. Hatakeyama); yosuke.gjdt.yurikusa@misumi.co.jp (Y. Yurikusa)
0009-0000-1465-8522 (T. Motegi); 0000-0002-9351-0901 (M. P. Kato)

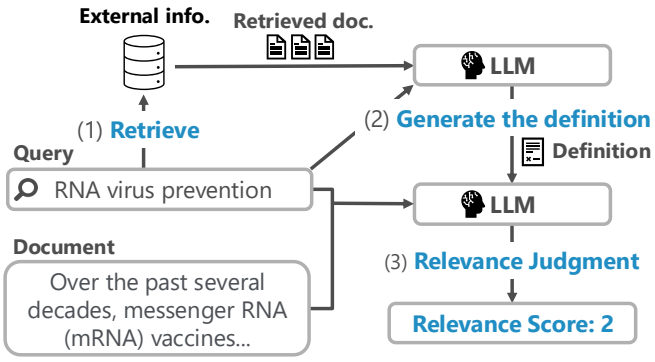


Figure 1: Overview of the Retrieval-Augmented Relevance Judgment (RAJ).

related to the query and uses an LLM to generate a definition sentence for the query based on those documents. A definition sentence is additional text describing what the query means. The generated definition sentence is then provided as context to the LLM for making the final relevance judgment. This approach aims to enhance accuracy by supplementing the LLM's limited domain-specific knowledge.

This study addresses the following research questions (RQs) through evaluation and analysis of RAJ: (**RQ1**) **Does retrieval augmentation improve the accuracy of LLM-based relevance judgments?** Although LLMs are trained on a vast amount of general knowledge, they may lack information specific to specialized domains. We examine whether providing external knowledge through retrieval enables more accurate judgments. (**RQ2**) **How do the content and format of information provided to an LLM affect the relevance judgment accuracy?** Because an LLM makes relevance judgments based on the information it receives, the content and format of that input are expected to influence its decisions. Therefore, we manipulate both the content and the format of the input information and evaluate how each affects the relevance judgment accuracy. (**RQ3**) **Does RAJ improve the relevance judgment accuracy particularly in specialized domains?** RAJ is expected to be particularly effective in domains where the LLM's pre-trained knowledge is insufficient. To verify this, we classify queries and documents into three levels of domain expertise using an LLM, and analyze how differences in expertise affect the judgment accuracy.

To evaluate RAJ, we conducted experiments on four datasets: TREC Robust 2004, TREC-COVID, NFCorpus, and a dataset from MISUMI, a Japanese e-commerce site for industrial parts. For RQ1, we compared RAJ to a baseline method that performs relevance judgment without any definition input. Judgments were evaluated by measuring agreement with human annotations. Results show that RAJ consistently achieved higher agreement than the baseline, particularly in specialized domains. Notably, RAJ improved recall for non-relevant documents in these domains. For RQ2, we tested three conditions to assess how input variations affect performance: providing a random definition, directly providing retrieved documents, and using an LLM-generated definition without retrieval. The results show that performance dropped in the first two conditions, suggesting the importance of supplying relevant and well-structured external knowledge. For RQ3, we used an LLM to classify documents by domain specificity and examined performance by domain type. We observed that COVID and NFCorpus datasets, which contain more highly specialized content, saw greater accuracy improvements compared to the general-domain Robust dataset.

The contributions of this paper are summarized as follows. First, we propose a relevance judgment method for specialized domains using retrieval augmentation. Second, we demonstrate the effectiveness of supplementing LLMs with web-searched external knowledge during relevance judgment. Third, we show that improving accuracy in relevance judgment requires not only supplying external knowledge but also properly complementing it with task-relevant information, such as definition sentences related to the query. Finally, we provide evidence that RAJ improves accuracy more substantially in highly specialized domains.

2. Related Work

2.1. Relevance Judgment Using LLMs

Recent studies have actively explored the use of LLMs for relevance judgment, demonstrating that LLMs can achieve high agreement with human annotations [10, 11, 12]. Many techniques have been proposed to enhance the accuracy of LLM-based relevance judgments. Thomas et al. investigated effective prompting strategies for instructing LLMs to perform relevance judgments [10]. There have also been domain-specific approaches proposed for high-expertise domains such as medicine and engineering. These domains often require careful adaptation to allow LLMs to perform relevance judgment at a quality comparable to human experts [7, 8]. However, to the best of our knowledge, these domain-specific methods are typically tailored to a single domain, and no existing methods have been proposed that generalize across multiple specialized domains. Studies have also examined the reliability and validity of using LLMs for relevance judgment [13, 14, 15]. Abbasiantaeb et al. observed that LLMs tend to assign higher relevance scores than humans [12]. Alaofi et al. investigated vulnerabilities in LLM-based relevance judgments, showing that LLMs are more likely to misjudge non-relevant documents as relevant when those documents contain surface-level term overlaps with the query [16]. Upadhyay et al. demonstrated that, at the run level, there is a high correlation between evaluation results based on LLM-derived relevance judgments and those based on human judgments. However, they also pointed out inconsistencies in the quality of topic-level judgments [17].

In this study, we propose using retrieval augmentation to improve the performance of LLM-based relevance judgment across multiple specialized domains.

2.2. Retrieval Augmentation

Retrieval augmentation has been shown to be effective in improving the performance of machine learning models. Zamani et al. proposed a comprehensive framework for integrating retrieval mechanisms into machine learning models, arguing that decoupling inference from memory through retrieval can enhance generalization, scalability, robustness, and interpretability [18]. Lewis et al. introduced Retrieval-Augmented Generation (RAG) [9], a method that provides retrieved documents to a language model during answer generation. By conditioning generation on retrieved content, RAG reduces hallucination and produces more factually accurate responses. Additionally, because the retrieval index can be updated independently of model parameters, responses can be refreshed without retraining the model. Thanks to these advantages, retrieval augmentation has been applied to a wide range of tasks. In question answering, it has been used to retrieve relevant documents based on the query, which are then used to generate more accurate answers [18, 19].

Beyond QA, retrieval augmentation has also been applied to intent classification [20], dataset generation [21], personalization [22], and recommendation[23]. However, to the best of our knowledge, it has not yet been applied to relevance judgment.

3. Methodology

3.1. Overview of RAJ

As the domain specificity of the corpus used for relevance judgment increases, the knowledge an LLM acquires from pre-training alone may be insufficient for accurate judgment. In highly specialized domains, providing definitions or explanations of the given queries can be helpful. Thomas et al. [10] showed that detailed query descriptions improve LLM-based relevance judgments. Motivated by these findings, we propose RAJ, a method that retrieves information related to a given query, reformats it into concise definition sentences, and provides these sentences to the LLM during relevance judgment. A definition sentence is additional text describing what the query means. By incorporating these definition sentences, RAJ aims to supplement missing domain knowledge and thereby improve judgment accuracy.

An overview of RAJ is shown in Figure 1. We first conduct a search, as indicated by (1) in the figure, using the query whose relevance is to be judged. The retrieved documents are then used to generate definition sentences. Next, as shown in (2), the retrieved documents and the target query are provided to the LLM, which produces definition sentences for the query. Supplying these sentences to the LLM during relevance judgment supplements the model's missing domain knowledge. Finally, as illustrated in (3), the LLM receives the target query, the candidate document, and the generated definition sentences, and returns a relevance judgment. In this study, we design a prompt for relevance judgment based on the prompt proposed by Thomas et al. [10], which has been shown to be effective for this task. We extend it to incorporate the generated definitions. The prompts and experimental settings can be found in our code repository[1]. We then compare two methods that differ in their retrieval targets for generating definition sentences. **RAJ (query)** : A method that retrieves documents using the relevance judgment query and generates a definition sentence from the retrieved information. **RAJ (entity)** : A method that extracts named entities from the query, retrieves documents for each entity, and generates a corresponding definition sentence for each.

3.2. RAJ (query)

In RAJ (query), documents are retrieved using the relevance judgment query, and definition sentences are generated from the retrieved results for use in relevance judgment. The prompt used for generating the definition sentences and sample outputs can be found in our repository[1]. By leveraging the contextual information available from the entire query during definition generation, it is expected that the LLM can better understand domain-specific terms and concepts, thereby enabling more accurate judgments. The details of RAJ (query) are shown in Algorithm 1. First, the function Retrieve(q) is used to obtain a set of relevant documents R_q based on the query q. Then, the function Generate(R_q) generates a definition sentence s_q from R_q. Subsequently, the function Judge(q, d, s_q) performs relevance judgment using the query q, document d, and definition sentence s_q, producing a label j. In RAJ (query), an additional API call is required for generating a definition sentence for each query. Consequently, both the number of API calls and the total number of generated tokens increase, leading to higher overall inference cost.

3.3. RAJ (entity)

In RAJ (entity), named entities are extracted from the relevance judgment query, and definition sentences are generated for each entity for use in relevance judgment. The prompt used for generating the definition sentences and sample outputs can be found in our repository[1]. This method aims to comprehensively supplement knowledge corresponding to multiple domain-specific elements within the query, which is expected to result in more accurate judgments. The details of RAJ (entity) are presented in Algorithm 2. First, the function NER(q) extracts a set of named entities E_q from the query q. For each named entity $e \in E_q$, the function Retrieve(e) is used to retrieve a set of related documents R_e. Next, the function Generate(R_e) generates a definition sentence s_e based on the retrieved documents, which is then added to the set S_e. Finally, the function Judge(q, d, S_e) performs the relevance judgment based on the query q, document d, and the set of generated definition sentences S_e, resulting in the label j. In RAJ (entity), an additional API call is required for generating a definition sentence for each named entity extracted from the query, so that the number of API calls is equal to the number of extracted entities. Consequently, both the number of API calls and token consumption increase proportionally with the number of entities, meaning that queries containing more named entities tend to incur higher costs than RAJ (query).

[1] https://github.com/kasys-lab/RAJ

Algorithm 1 RAJ (query)

Input: Query q, Document d
Output: Relevance label j

$R_q \leftarrow$ Retrieve(q) ▷ Retrieve with the query
$s_q \leftarrow$ Generate(R_q) ▷ Generate a definition
$j \leftarrow$ Judge($q, d, \{s_q\}$) ▷ Judge with the def.
return j

Algorithm 2 RAJ (entity)

Input: Query q, Document d
Output: Relevance label j

$E_q \leftarrow$ NER(q) ▷ Extract named entities (NEs)
$S_e \leftarrow \emptyset$
for $e \in E_q$ **do**
    $R_e \leftarrow$ Retrieve(e) ▷ Retrieve with the NE
    $s_e \leftarrow$ Generate(R_e)
    $S_e \leftarrow S_e \cup \{s_e\}$
end for
$j \leftarrow$ Judge(q, d, S_e) ▷ Judge with the def.
return j

Table 1
Datasets and their statistics.

Dataset	Domain	Relevant	Partially Relevant	Non-Relevant	Total
Robust	General	900	900	1,800	3,600
COVID	Medical	1,000	1,000	2,000	4,000
NFCorpus	Medical	500	500	1,000	2,000
MISUMI	Industrial	450	450	900	1,800

4. Experiments and Results

4.1. Experimental Setup

Datasets In this study, we evaluate RAJ on four datasets: TREC Robust 2004 (Robust), a news collection spanning diverse topics [24]; TREC-COVID (COVID), which contains scientific papers on COVID-19 [25]; NFCorpus, a medical corpus [26]; and a proprietary dataset from the Japanese industrial-parts e-commerce site MISUMI. Robust is a general-purpose corpus, whereas the remaining three focus on specialized domains. Because NFCorpus lacks explicit non-relevant labels, we retrieve the top 100 documents for each query using BM25 and treat any document not judged relevant as non-relevant. The MISUMI dataset, being private, has never been used to train public models, allowing us to measure the performance without concerns about data leakage into LLMs. MISUMI contains queries, product information, and human relevance judgments in five categories, namely, *Exact Match*, *Near Equivalent*, *Related*, *Material Match*, and *Unrelated*. In our experiments, Exact Match and Near Equivalent are treated as relevant, Related and Material Match as partially relevant, and Unrelated as non-relevant. For each dataset, we randomly sampled relevant, partially relevant, and non-relevant documents in a 1:1:2 ratio to ensure equal numbers of positive samples (relevant and partially relevant) and negative samples (non-relevant) for the binary classification task described later. Dataset statistics are summarized in Table 1.

Implementation and Evaluation of RAJ For named entity extraction in RAJ (entity), we used GPT-4 via the spaCy-llm NLP library. Document retrieval was performed using the Bing Web Search API, from which we collected the top 10 URLs. We then extracted the main text from each URL using the text extraction library trafilatura [27]. Among the retrieved documents, we used the top 3 with successful text extraction for generating definition sentences. Definition generation was conducted using GPT-4 (gpt-4-0613). Since GPT-4 has a context window of 8,192 tokens, and it was not feasible to input the entire contents of three documents, we used only the first 2,500 tokens from each document

Table 2
Example of a definition sentence generated for the query "coronavirus and ACE inhibitors"

Method	Generated Definition Sentence
RAJ (query)	ACE inhibitors and angiotensin receptor blockers (ARBs) are medications often used in the treatment of hypertension and heart failure, which have been studied for their potential impact on COVID-19 disease severity. Some studies suggest that these drugs may be associated with reduced risks of severe COVID-19 disease, while others indicate no significant increase in risk. The relationship between these medications and COVID-19 is complex due to the role of the renin-angiotensin-aldosterone system (RAAS) and its interaction with SARS-CoV-2, the virus causing COVID-19.
RAJ (entity)	Coronavirus refers to a group of related RNA viruses that cause diseases in mammals and birds, including humans, where they can cause respiratory tract infections ranging from mild to lethal, with severe forms known as SARS, MERS and COVID-19. ACE inhibitors, short for Angiotensin-converting enzyme inhibitors, are a class of medications primarily used to treat heart and kidney conditions by blocking the production of angiotensin II, a substance that narrows blood vessels and increases blood pressure, thereby allowing blood vessels to relax and dilate, reducing both blood and kidney pressure.
GAJ (query)	ACE inhibitors are a type of medication used to treat high blood pressure and heart conditions, and there has been research into their potential effects on the severity of symptoms in patients with the coronavirus, a highly infectious respiratory disease caused by SARS-CoV-2.

for definition generation. An example of the generated definition sentence is shown in Table 2. In RAJ (entity), definition sentences were generated only when named entities could be successfully extracted from the query. Definition sentences were generated for 76.6%, 100%, 80.7%, and 76.2% of query-document pairs in Robust, COVID, NFCorpus, and MISUMI, respectively. For relevance judgment, we used GPT-4 and GPT-3.5-turbo[2]. The generation parameters followed those of Thomas et al. [10], with $\mathrm{top\_p} = 1.0$, $\mathrm{frequency\_penalty} = 0.5$, $\mathrm{presence\_penalty} = 0$, and $\mathrm{temperature} = 0$. The prompt instructed the LLM to output 2 for highly relevant, 1 for relevant, and 0 for non-relevant cases. If RAJ (entity) failed to generate a definition sentence for a given query, relevance judgment was performed without any additional information. As a baseline, we used a relevance-judgment prompt without incorporating the generated definitions. The prompt used for relevance judgment can be found in our repository[1].

We evaluated each method by measuring agreement with human relevance judgments using Accuracy, Cohen's κ[28], and Krippendorff's α[29] for ordinal data. Evaluation was performed using two approaches: binary classification and ordinal agreement. For binary classification, labels 2 and 1 were treated as relevant, and 0 as non-relevant.

4.2. RQ1. Does retrieval augmentation improve the accuracy of LLM-based relevance judgments?

Table 3 shows the agreement between the LLMs' relevance judgments and human annotations. In the table, bold values indicate the best performance for each metric, and underlined values indicate statistically significant improvements in accuracy over the baseline ($\alpha = 0.05$). To assess significance, we conducted McNemar's test on the accuracy between the baseline and RAJ, followed by Holm correction across datasets and GPT versions. With GPT-4, RAJ (query) achieved the best performance across all datasets in terms of Accuracy, Cohen's κ, and Krippendorff's α. In particular, since judgment accuracy improved in highly specialized domains, these results suggest that the more specialized

[2]GPT-4-0613 and GPT-3.5-turbo-0613 were used for RQ1, while GPT-4-turbo-2024-04-09 and GPT-3.5-turbo-0125 were used for RQ2, due to the availability in Azure OpenAI Service.

Table 3
Evaluation results of relevance judgment accuracy for each method. Acc.: Accuracy, κ: Cohen's Kappa, α: Krippendorff's α. The models used are GPT-4: GPT-4-0613, GPT-3.5-turbo: GPT-3.5-turbo-0613. Bold indicates the best result for each metric, and underlining indicates a statistically significant difference in accuracy compared to the baseline (α = 0.05).

(a) Evaluation results for binary classification

Model	Method	Robust			COVID			NFCorpus			MISUMI			Avg.		
		Acc.	κ	α	Acc.	κ	α	Acc.	κ	α	Acc.	κ	α	Acc.	κ	α
GPT-4	Baseline	.873	.747	.746	.733	.465	.454	.629	.258	.233	.616	.232	.152	.713	.425	.396
	RAJ (query)	**.881**	**.762**	**.762**	**.744**	**.489**	**.480**	**.650**	**.300**	**.289**	**.706**	**.411**	**.408**	**.745**	**.490**	**.485**
	RAJ (entity)	.878	.757	.756	.710	.420	.402	.625	.249	.233	.665	.330	.311	.720	.439	.426
GPT-3.5-turbo	Baseline	**.737**	**.473**	**.458**	.641	.281	.202	.573	.145	.141	.578	.156	.089	.632	.264	.223
	RAJ (query)	.682	.364	.306	**.724**	**.448**	**.503**	**.638**	**.275**	**.251**	**.637**	**.274**	**.265**	**.670**	**.340**	**.317**
	RAJ (entity)	.714	.429	.400	.654	.308	.258	.596	.192	.188	.603	.207	.207	.642	.284	.263

(b) Evaluation results for three-class classification

Model	Method	Robust			COVID			NFCorpus			MISUMI			Avg.		
		Acc.	κ	α	Acc.	κ	α	Acc.	κ	α	Acc.	κ	α	Acc.	κ	α
GPT-4	Baseline	.693	.727	.740	.570	.538	.525	.536	.405	.347	.393	.232	.103	.548	.475	.429
	RAJ (query)	.698	**.738**	**.759**	**.591**	**.579**	**.570**	**.555**	**.475**	**.425**	**.536**	**.441**	**.444**	**.595**	**.558**	**.549**
	RAJ (entity)	**.702**	.729	.746	.540	.516	.496	.535	.419	.361	.489	.367	.349	.567	.508	.488
GPT-3.5-turbo	Baseline	**.631**	**.520**	**.510**	.418	.278	.138	.467	.249	.228	.368	.161	.067	.471	.302	.236
	RAJ (query)	.603	.424	.361	**.572**	**.503**	**.502**	**.572**	**.443**	**.378**	**.518**	**.281**	**.276**	**.566**	**.413**	**.379**
	RAJ (entity)	**.631**	.494	.461	.453	.332	.264	.514	.343	.309	.469	.251	.248	.517	.355	.320

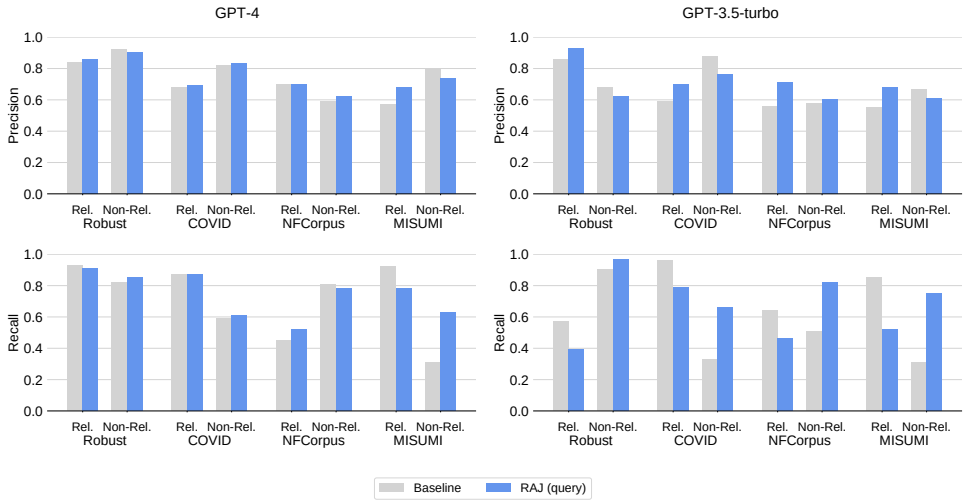


Figure 2: Precision and Recall of the baseline and RAJ (query).

the target dataset, the greater the potential of RAJ to enhance judgment accuracy. In contrast, no significant difference was observed on the Robust dataset. Since the baseline already achieved high accuracy on Robust, it is likely that the additional information introduced by RAJ was redundant and did not contribute to improved performance. Even with GPT-3.5-turbo, RAJ (query) outperformed the baseline on every dataset except Robust. Compared to RAJ (query), RAJ (entity) generally showed lower agreement with human judgments. Extracting named entities before generating definitions sometimes led to a loss of query context, resulting in definitions that were misaligned with the intent of the original

Table 4
Evaluation results of relevance judgment accuracy under each condition. Acc.: Accuracy, κ: Cohen's Kappa, α: Krippendorff's α. The models used are GPT-4-turbo (GPT-4-turbo-2024-04-09) and GPT-3.5-turbo (GPT-3.5-turbo-0125). Bold indicates the best result for each metric, and underlining indicates a statistically significant difference in accuracy compared to the baseline (α = 0.05).

(a) Evaluation results for binary classification

Model	Method	Robust			COVID			NFCorpus			Avg.		
		Acc.	κ	α	Acc.	κ	α	Acc.	κ	α	Acc.	κ	α
GPT-4-turbo	Baseline	.864	.727	.727	.756	.512	.513	.612	.224	.175	.744	.488	.472
	RAJ (query)	.859	.717	.717	**.763**	**.526**	**.525**	.646	.293	**.255**	**.756**	**.512**	**.499**
	(1) w/ random	.843	.686	.686	.762	.524	.524	.607	.213	.157	.737	.474	.456
	(2) w/o sum.	.775	.549	.536	.724	.447	.435	**.647**	**.294**	.233	.715	.430	.402
	(3) GAJ (query)	**.873**	**.746**	**.746**	.758	.516	.516	.637	.274	.232	**.756**	**.512**	.498
GPT-3.5-turbo	Baseline	**.793**	**.586**	**.586**	.632	.264	.175	.573	.146	.144	.666	.332	.301
	RAJ (query)	.741	.482	.462	.721	.443	.442	**.626**	**.252**	.217	.696	.392	.374
	(1) w/ random	.715	.429	.407	.684	.369	.362	.573	.146	.128	.657	.315	.299
	(2) w/o sum.	.689	.378	.377	.624	.249	.198	.612	.223	**.219**	.642	.283	.265
	(3) GAJ (query)	.761	.523	.510	**.723**	**.446**	**.445**	.615	.229	.183	**.700**	**.399**	**.379**

(b) Evaluation results for three-class classification

Model	Method	Robust			COVID			NFCorpus			Avg.		
		Acc.	κ	α	Acc.	κ	α	Acc.	κ	α	Acc.	κ	α
GPT-4-turbo	Baseline	.697	.706	**.725**	.620	.588	.586	.532	.346	.264	.616	.546	.525
	RAJ (query)	**.698**	700	723	**.642**	**.597**	**.591**	.567	.430	**.355**	**.636**	**.576**	**.556**
	(1) w/ random	.682	.679	.694	.631	.587	.586	.548	.373	.277	.620	.546	.519
	(2) w/o sum.	.640	.549	.565	.610	.498	.482	**.572**	**.445**	.342	.607	.503	.464
	(3) GAJ (query)	.697	**.746**	.716	.624	.579	.579	.557	.398	.322	.626	.564	.547
GPT-3.5-turbo	Baseline	.629	**.573**	**.584**	.397	.207	−.013	.467	.235	.208	.498	.338	.260
	RAJ (query)	**.644**	.561	.541	**.576**	**.474**	**.476**	**.588**	**.429**	**.354**	**.603**	**.488**	**.457**
	(1) w/ random	.614	.493	.476	.514	.376	.356	.516	.298	.253	.548	.389	.362
	(2) w/o sum.	.549	.377	.384	.425	.242	.144	.496	.326	.303	.490	.315	.277
	(3) GAJ (query)	**.644**	**.573**	.570	.559	.443	.441	.573	.375	.301	.592	.464	.437

query. This contextual mismatch is likely a key factor in the reduced performance. In addition, the failure of definition generation is another possible reason. As reported in Section 4.1, RAJ (entity) failed to generate definition sentences for some queries, so we inserted a null value in the definition field for those queries and then carried out the relevance judgment.

Figure 2 shows the precision and recall for each relevance label of the baseline and RAJ (query). Compared to the baseline, RAJ (query) achieves higher recall for non-relevant documents while maintaining a relatively similar level of precision. Only in the case of GPT-4 on the NFCorpus dataset does recall for relevant documents increase compared to the baseline. According to the study by Alaofi et al. [16], LLMs tend to predict documents as relevant if query terms appear in them. However, this tendency appears to be mitigated in RAJ. By referring to the definition sentence, even if GPT lacks domain knowledge, the criteria for relevance judgment become clearer, making it easier to identify non-relevant documents. In the case of GPT-3.5-turbo, there is a trend of decreased recall for relevant documents and increased recall for non-relevant ones. Compared to GPT-4, GPT-3.5-turbo appears to rely more on the definition sentence when making relevance judgments.

In summary, we demonstrated that RAJ (query) consistently outperforms the baseline on specialized datasets. On the other hand, RAJ (entity) exhibited lower performance due to incomplete definition generation and loss of contextual information. Furthermore, RAJ (query) showed a tendency to improve recall for non-relevant documents, likely due to the disambiguation support provided by the generated

definitions.

4.3. RQ2. How do the content and format of information provided to an LLM affect the relevance judgment accuracy?

To address RQ2, we investigated how differences in the input information provided to LLMs influence the accuracy of relevance judgments. Specifically, we compared performance under the following three conditions. (1) Relevance judgment using randomly assigned definition sentences. (2) Relevance judgment using full documents retrieved from web search without summarization. (3) Relevance judgment using definition sentences generated by an LLM without retrieval augmentation.

In condition (1), to test whether simply supplying a definition sentence boosts the accuracy, we randomly selected a definition sentence from the same dataset that belonged to a different query, appended it, and then performed relevance judgment. In condition (2), we tested whether summarizing retrieved information into a definition sentence improves the accuracy by skipping the summary step. We fed the full text of the top document returned by the Bing Web Search API directly into the relevance judgment model. If the document exceeded the model token limit we truncated it to the first 10,000 tokens. In condition (3), we assessed whether generating definition sentences from retrieved external documents actually improves the accuracy by instead omitting retrieval, generating the definitions, and then performing relevance judgment. This approach, called Generation-Augmented Relevance Judgment (GAJ), creates a definition sentence using only the target query together with a single example. The example is the same manually written definition sample used in RAJ and is included in the prompt. The definition sentence generated by GAJ is shown in Table 2. The prompt used for generating the definition sentences and sample outputs can be found in our repository[1]. Definition sentences were generated with GPT-4, and relevance judgments were performed with GPT-4-turbo and GPT-3.5-turbo. The MISUMI dataset was excluded from this experiment due to limited access at the time of execution.

Additionally, we hypothesized that the more similar a definition sentence is to highly relevant documents, the more effectively it serves as a criterion for determining relevance, thereby improving the relevance judgment accuracy. To test this hypothesis, we generated embedding representations for the definition sentences and the highly relevant documents (relevance = 2) and computed their cosine similarity. When a single definition sentence corresponded to multiple documents, we calculated the similarity for each pair and used the average value. Embeddings were generated using the multilingual-e5-large model[3].

Table 4 presents the evaluation results for the relevance judgment accuracy under different input configurations to the LLM. In the table, bold values represent the best performance for each metric, and underlined values indicate statistically significant improvements in accuracy over the baseline ($\alpha = 0.05$). Following Section 4.2, we conducted McNemar's test on the accuracy between the baseline and each method, and applied Holm correction to confirm statistical significance. In condition (1), which used randomly assigned definition sentences, we observed little to no improvement in relevance judgment accuracy. The only exception was with GPT-4-turbo on the COVID dataset, where performance improved slightly over the baseline. This may be attributed to the fact that most queries in COVID contain the term "coronavirus", making even unrelated definitions somewhat informative. Similarly, with GPT-3.5-turbo, improvements were observed on the COVID and NFCorpus datasets, suggesting that providing any definition sentence may influence relevance judgment in lighter-weight models. However, compared with RAJ (query), judgment accuracy decreased across all datasets. This result shows that adding external knowledge does not automatically improve accuracy and that providing knowledge relevant to relevance judgment is essential. For condition (2), which used unsummarized retrieved documents, performance generally declined. An exception was observed for GPT-4-turbo on the NFCorpus dataset, where the accuracy was improved. Compared with RAJ (query), judgment accuracy decreased in all cases except for NFCorpus with GPT-4-turbo. This result suggests that using retrieved documents without summarization may introduce irrelevant information that hinders accurate

[3] https://huggingface.co/intfloat/multilingual-e5-large

Table 5
Correlation coefficients between accuracy and similarity scores. Bold indicates the best result for each metric, and underlining indicates a statistically significant correlation (α = 0.05).

Model	Method	Robust	COVID	NFCorpus
GPT-4-turbo	RAJ (query)	**.313**	.267	.360
	RAJ (entity)	.097	.242	.227
	GAJ (query)	.288	**.458**	**.374**
GPT-3.5-turbo	RAJ (query)	.398	.448	.413
	RAJ (entity)	.171	.315	.240
	GAJ (query)	**.472**	**.506**	**.435**

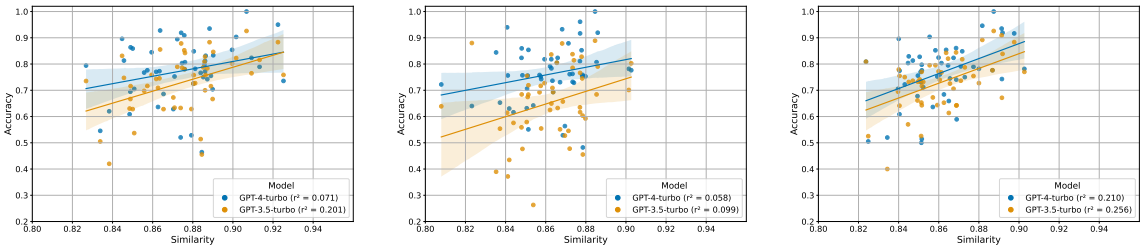
relevance judgment. In condition (3), which used LLM-generated definitions without retrieval (GAJ), performance generally improved, except for GPT-3.5-turbo on the Robust dataset. As Robust is a general-domain corpus, the generated definitions may have lacked specificity, resulting in ambiguous input and degraded performance. Although GAJ (query) outperformed the baseline, it remained inferior to RAJ (query), confirming that retrieval-augmented knowledge is particularly beneficial in specialized domains.

Table 5 shows, for each method, the accuracy under the binary classification setting and the correlation coefficients between the similarity of the definition sentences and the highly relevant documents. Bold values denote the highest correlation for each model, and underlined values indicate statistically significant correlation ($\alpha = 0.05$). We tested for statistical independence between the accuracy and similarity using correlation tests, with Holm correction applied across datasets and models. The correlation was higher for GPT-3.5-turbo than for GPT-4-turbo, suggesting that GPT-3.5-turbo relevance judgments were more influenced by the provided definition sentences. Figure 3 presents a scatter plot showing the relationship between the accuracy and the cosine similarity of definitions and highly relevant documents for the COVID dataset. Across all methods we observed a positive trend. The greater the similarity between the definition and the highly relevant documents, the higher the judgment accuracy. This likely occurs because the generated definition sentences act as pseudo-relevant documents, which clarifies what is non-relevant and improves the identification of non-relevant documents. Zhang et al. report that, within their Retrieval-Augmented Verification (RAV) framework for fact verification, elevating sentences that are closer to the gold evidence to higher ranks in the retrieval list clearly improves the verification accuracy [30].

In summary, we confirmed that simply providing external knowledge is insufficient; it is essential to supplement information that directly supports the judgment. Approaches that provide unrelated definitions or unfiltered web-searched documents can reduce the accuracy due to noise and inconsistency. The comparison between GAJ (query) and RAJ (query) suggests that as dataset specialization increases, the knowledge within the LLM alone becomes inadequate, and retrieval-based augmentation improves performance. We also found that higher similarity between definition sentences and highly relevant documents correlates with better accuracy, highlighting the importance of quality and contextual fit in relevance judgment.

4.4. RQ3. Does RAJ improve the relevance judgment accuracy particularly in specialized domains?

To answer RQ3, we conducted expertise annotation for each query-document pair subject to relevance judgment using an LLM. Based on the relevance judgment prompt proposed by Thomas et al.[10], we designed a prompt that instructed the LLM to classify both queries and documents into three levels of expertise: Highly specialized, Moderately specialized, and Non-specialized. The prompt used for expertise annotation can be found in our repository[1]. We used three datasets: Robust, COVID, and NFCorpus, and the model employed was GPT-4-turbo. Table 6 shows the classification results of queries and documents into expertise levels by GPT-4-turbo. For queries, across all datasets, more than 80% were



(a) RAJ (query) (b) RAJ (entity) (c) GAJ (query)

Figure 3: Scatter plot of similarity between definition sentences and highly relevant documents vs. accuracy in COVID.

Table 6
Number and percentage (%) of queries and documents subject to relevance judgment classified into three levels of specialization by GPT-4-turbo

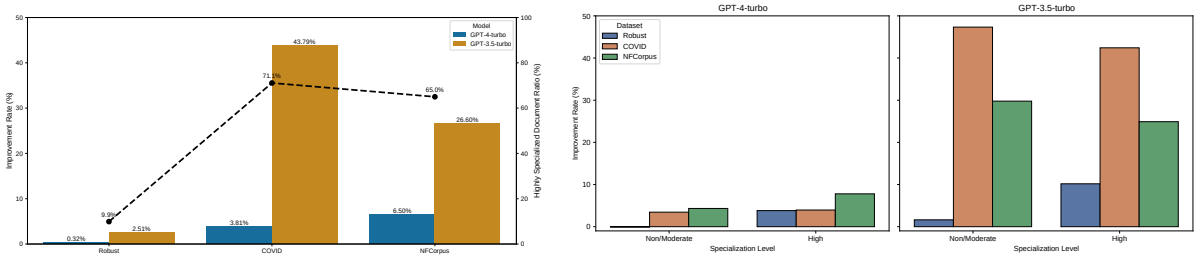
Dataset	Category	High		Moderate		Non	
		Num.	%	Num.	%	Num.	%
Robust	query	5	2.0	132	53.2	111	44.8
	doc	353	9.9	1811	51.0	1387	39.1
COVID	query	6	12.0	35	70.0	9	18.0
	doc	2605	71.1	1034	28.2	24	0.7
NFCorpus	query	13	5.0	114	43.7	134	51.3
	doc	973	65.0	519	34.6	6	0.4

classified as either Moderately specialized or Non-specialized. In particular, the majority of NFCorpus queries were categorized as Non-specialized, likely due to their intentional avoidance of technical terminology, despite being from the medical domain. For documents, in the Robust dataset, a large portion was classified as Moderately specialized, whereas in COVID and NFCorpus, many documents were labeled as Highly specialized. Furthermore, in COVID and NFCorpus, only a small number of documents were categorized as Non-specialized. These results indicate that GPT-4-turbo recognizes COVID and NFCorpus as corpora with higher domain specificity.

Figure 4a presents a line graph showing the proportion of documents classified as Highly specialized for each dataset, and a bar chart depicting the accuracy improvement of RAJ (query) over the baseline on a per-document basis. The improvement rate was calculated by taking the mean difference in three-class classification accuracy between the baseline and RAJ (query), and dividing it by the mean baseline accuracy. For GPT-3.5-turbo, we observed a trend that the higher the proportion of highly specialized documents, the greater the improvement in Accuracy with RAJ (query). Similarly, for GPT-4, the improvement rate was higher for COVID and NFCorpus than for Robust. These results suggest that RAJ (query) tends to improve judgment accuracy as the proportion of specialized documents increases.

Figure 4b illustrates the accuracy improvement of RAJ (query), grouped by whether the document was classified as Highly specialized or not. With GPT-4-turbo, documents with higher expertise levels showed greater improvements. In the Robust dataset, no improvement was observed for documents with lower expertise, while a 3.8% improvement was observed for highly specialized ones. In contrast, for GPT-3.5-turbo, no clear trend was observed that more specialized documents lead to higher improvement rates, except for Robust.

In summary, we demonstrated that RAJ (query) achieves significant accuracy improvements over the baseline, especially in highly specialized domains such as COVID and NFCorpus. The domain specificity annotations provided by GPT-4-turbo confirmed that many documents in these datasets require advanced domain knowledge. In such cases, definition sentences generated via retrieval augmentation effectively complement the LLM's internal knowledge and contribute to improved relevance judgment accuracy.



(a) Proportion of highly specialized documents and accuracy improvement of RAJ (query).

(b) Accuracy improvement of RAJ (query) by document specialization level.

Figure 4: The improvement rate in accuracy of RAJ (query) over the baseline from the perspective of document specialization.

5. Conclusion and Future Work

In this paper, we proposed Retrieval-Augmented Relevance Judgment (RAJ), a method that leverages LLMs for relevance judgment in specialized domains such as medicine and manufacturing. To address missing domain knowledge, RAJ supplements queries with retrieved definitions and was evaluated against human judgments on TREC Robust 2004, TREC-COVID, NFCorpus, and MISUMI. RQ1 showed that RAJ improves agreement with human judgments, with larger gains in specialized domains, mainly by increasing the recall of non-relevant documents. RQ2 revealed that irrelevant or noisy definitions reduce accuracy, highlighting the importance of high-quality, contextually appropriate knowledge. RQ3 confirmed that RAJ (query) improves judgment accuracy over the baseline, particularly in highly specialized domains such as COVID and NFCorpus.

Future work includes verifying the accuracy of relevance judgments when providing retrieval-based information about documents rather than queries. In addition, while this paper evaluated performance based on the agreement of relevance judgment labels, another important direction for future work is to assess agreement based on the consistency of retrieval model rankings.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP23K28090.

References

[1] H. A. Rahmani, C. Siro, M. Aliannejadi, N. Craswell, C. L. A. Clarke, G. Faggioli, B. Mitra, P. Thomas, E. Yilmaz, Llm4eval: Large language model for evaluation in ir, in: SIGIR, 2024, p. 3040–3043.

[2] G. Faggioli, L. Dietz, C. L. A. Clarke, G. Demartini, M. Hagen, C. Hauff, N. Kando, E. Kanoulas, M. Potthast, B. Stein, H. Wachsmuth, Perspectives on large language models for relevance judgment, in: ICTIR, 2023, p. 39–50.

[3] G. Xiong, Q. Jin, Z. Lu, A. Zhang, Benchmarking retrieval-augmented generation for medicine, in: ACL, 2024, pp. 6233–6251.

[4] Y. Zhao, P. Singh, H. Bhathena, B. Ramos, A. Joshi, S. Gadiyaram, S. Sharma, Optimizing LLM based retrieval augmented generation pipelines in the financial domain, in: ACL, 2024, pp. 279–294.

[5] W. Qin, Z. Cao, W. Yu, Z. Si, S. Chen, J. Xu, Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach, in: SIGIR, 2024, p. 2210–2220.

[6] B. Soviero, D. Kuhn, A. Salle, V. P. Moreira, Chatgpt goes shopping: Llms can predict relevance in ecommerce search, in: ECIR, 2024, p. 3–11.

[7] S. Ma, C. Chen, Q. Chu, J. Mao, Leveraging large language models for relevance judgments in legal case retrieval, arXiv preprint arXiv:2403.18405 (2024).

[8] N. Mehrdad, H. Mohapatra, M. Bagdouri, P. Chandran, A. Magnani, X. Cai, A. Puthenputhussery, S. Yadav, T. Lee, C. Zhai, et al., Large language models for relevance judgment in product search, arXiv preprint arXiv:2406.00247 (2024).
[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: NeurIPS, 2020.
[10] P. Thomas, S. Spielman, N. Craswell, B. Mitra, Large language models can accurately predict searcher preferences, in: SIGIR, 2024, p. 1930–1940.
[11] S. Upadhyay, E. Kamalloo, J. Lin, Llms can patch up missing relevance judgments in evaluation, arXiv preprint arXiv:2405.04727 (2024).
[12] Z. Abbasiantaeb, C. Meng, L. Azzopardi, M. Aliannejadi, Can we use large language models to fill relevance judgment holes?, arXiv preprint arXiv:2405.05600 (2024).
[13] C. L. Clarke, L. Dietz, Llm-based relevance assessment still can't replace human relevance assessment, in: EVIA, 2025.
[14] I. Soboroff, Don't use llms to make relevance judgments, Information Retrieval Research (2025) 29–46.
[15] K. Balog, D. Metzler, Z. Qin, Rankers, judges, and assistants: Towards understanding the interplay of llms in information retrieval evaluation, in: SIGIR, 2025, pp. 3865–3875.
[16] M. Alaofi, P. Thomas, F. Scholer, M. Sanderson, Llms can be fooled into labelling a document as relevant: best café near me; this paper is perfectly relevant, in: SIGIR-AP, 2024, p. 32–41.
[17] S. Upadhyay, R. Pradeep, N. Thakur, D. Campos, N. Craswell, I. Soboroff, J. Lin, A large-scale study of relevance assessments with large language models using umbrela, in: ICTIR, 2025, p. 358–368.
[18] H. Zamani, F. Diaz, M. Dehghani, D. Metzler, M. Bendersky, Retrieval-enhanced machine learning, in: SIGIR, 2022, p. 2875–2886.
[19] G. Izacard, E. Grave, Leveraging passage retrieval with generative models for open domain question answering, in: EACL, 2021, pp. 874–880.
[20] J. Liu, T. Y. Keat, B. Fu, K. H. Lim, LARA: Linguistic-adaptive retrieval-augmentation for multi-turn intent classification, in: EMNLP, 2024, pp. 1096–1106.
[21] A. Divekar, G. Durrett, SynthesizRR: Generating diverse datasets with retrieval augmentation, in: EMNLP, 2024, pp. 19200–19227.
[22] A. Salemi, S. Kallumadi, H. Zamani, Optimization methods for personalizing large language models through retrieval augmentation, in: SIGIR, 2024, p. 752–762.
[23] J. Wu, C.-C. Chang, T. Yu, Z. He, J. Wang, Y. Hou, J. McAuley, Coral: Collaborative retrieval-augmented large language models improve long-tail recommendation, in: KDD, 2024, p. 3391–3401.
[24] E. Voorhees, Overview of the trec 2004 robust retrieval track, in: TREC, 2004.
[25] E. Voorhees, T. Alam, S. Bedrick, D. Demner-Fushman, W. R. Hersh, K. Lo, K. Roberts, I. Soboroff, L. L. Wang, Trec-covid: constructing a pandemic information retrieval test collection, in: ACM SIGIR Forum, volume 54, 2021, pp. 1–12.
[26] V. Boteva, D. Gholipour, A. Sokolov, S. Riezler, A full-text learning to rank dataset for medical information retrieval, in: ECIR, 2016, pp. 716–722.
[27] A. Barbaresi, Trafilatura: A web scraping library and command-line tool for text discovery and extraction, in: ACL, 2021, pp. 122–131.
[28] J. Cohen, A coefficient of agreement for nominal scales, Educational and psychological measurement 20 (1960) 37–46.
[29] K. Krippendorff, Computing krippendorff's alpha-reliability, 2011.
[30] L. Zheng, C. Li, X. Zhang, Y.-M. Shang, F. Huang, H. Jia, Evidence retrieval is almost all you need for fact verification, in: ACL, 2024, pp. 9274–9281.