Problem with existing RAG: Only looks at relevance.
Knowledge actually - Vary by source (forums vs gov sites), Evolve over time, Be complementary rather than directly answering, Contain conflicts.

POLYRAG: Integrating Polyviews into Retrieval-Augmented Generation for Medical Applications

Chunjing Gan Dan Yang Binbin Hu Ziqi Liu Yue Shen
Zhiqiang Zhang Jian Wang Jun Zhou[†]
Ant Group
jun.zhoujun@antgroup.com

Abstract

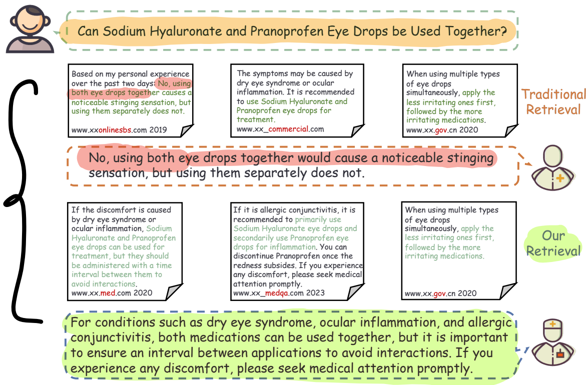
Large language models (LLMs) have become a disruptive force in the industry, introducing unprecedented capabilities in natural language processing, logical reasoning and so on. However, the challenges of knowledge updates and hallucination issues have limited the application of LLMs in medical scenarios, where retrieval-augmented generation (RAG) can offer significant assistance. Nevertheless, existing retrieve-then-read approaches generally digest the retrieved documents, without considering the timeliness, authoritativeness and commonality of retrieval. We argue that these approaches can be suboptimal, especially in real-world applications where information from different sources might conflict with each other and even information from the same source in different time scale might be different, and totally relying on this would deteriorate the performance of RAG approaches. We propose POLYRAG that carefully incorporate judges from different perspectives and finally integrate the polyviews for retrieval augmented generation in medical applications. Due to the scarcity of real-world benchmarks for evaluation, to bridge the gap we propose POLYEVAL, a benchmark consists of queries and documents collected from real-world medical scenarios (including medical policy, hospital & doctor inquiry and healthcare) with multiple tagging (*e.g.,* timeliness, authoritativeness) on them. Extensive experiments and analysis on POLYEVAL have demonstrated the superiority of POLYRAG[1].

Solely relying on relevance leads to hallucinations and errors

1 Introduction

Recently, large language models (LLMs) such as GPT4 (OpenAI, 2023), Llama3 (Grattafiori et al., 2024), Qwen (Yang et al., 2024), Deepseek-R1 (DeepSeek-AI et al., 2025) have become a disruptive force in the industry, which introduces marvelous capabilities in natural language processing (Mallen et al., 2023), logical reasoning (Patel et al., 2024), multi-modal processing (Zhang et al., 2024a) and so on. However, the heavy costs of knowledge updates (Shi et al., 2024a) and the long-standing hallucination issues (Gao et al., 2023a) have limited the application of LLMs in medical scenarios where incorrect answers may result in severe consequences, in this case retrieval-augmented generation (RAG) can be of help. Nevertheless, existing retrieve-then-read approaches generally directly digest the documents from the retrieval stages (Asai et al., 2024), without considering other perspectives such as timeliness, authoritativeness and commonality of retrieval.

Incomplete Response. Covers only one aspect.
Response generated only on Source 1. S2 and S3 contains other view of the knowledge.


Can Sodium Hyaluronate and Pranoprofen Eye Drops be Used Together?

Based on my personal experience over the past two days: No, using both eye drops together causes a noticeable stinging sensation, but using them separately does not.
www.xxonlinesbs.com 2019

The symptoms may be caused by dry eye syndrome or ocular inflammation. It is recommended to use Sodium Hyaluronate and Pranoprofen eye drops for treatment.
www.xx_commercial.com

When using multiple types of eye drops simultaneously, apply the less irritating ones first, followed by the more irritating medications.
www.xx.gov.cn 2020

No, using both eye drops together would cause a noticeable stinging sensation, but using them separately does not.

If the discomfort is caused by dry eye syndrome or ocular inflammation, Sodium Hyaluronate and Pranoprofen eye drops can be used for treatment, but they should be administered with a time interval between them to avoid interactions.
www.xx.med.com 2020

If it is allergic conjunctivitis, it is recommended to primarily use Sodium Hyaluronate eye drops and secondarily use Pranoprofen eye drops for inflammation. You can discontinue Pranoprofen once the redness subsides. If you experience any discomfort, please seek medical attention promptly.
www.xx_medqa.com 2023

When using multiple types of eye drops simultaneously, apply the less irritating ones first, followed by the more irritating medications.
www.xx.gov.cn 2020

For conditions such as dry eye syndrome, ocular inflammation, and allergic conjunctivitis, both medications can be used together, but it is important to ensure an interval between applications to avoid interactions. If you experience any discomfort, please seek medical attention promptly.

Figure 1: A toy example illustrating the difference between traditional retrieval and our retrieval strategy, where beyond relevance of a document, we also takes other perspectives such as its authoritativeness into consideration.

Complete and more accurate response.

Follow P. 2

Here, we argue that oftentimes these approaches can be suboptimal, especially in real-world applications (*e.g.,* medical applications) where not only information from different sources with respect to the same fact might conflict with each other but also information from the same source in different time scale might be different, and directly relying on them for generation would deteriorate the performance of RAG approaches. As the toy example shown in Figure 1, when a user types in

[1] We will release the data of POLYEVAL soon.

For example, a 2018 guideline might say a drug is safe during pregnancy, while a 2023 update says it's not. If a RAG model retrieves both without judging timeliness or source authority, it may produce incorrect answers.

the query “Can Sodium Hyaluronate and Pranoprofen Eye Drops be Used Together?”, a traditional RAG system would search and rank documents according to its relevance to the query (Shi et al., 2024a). Though the retrieved documents comes from non-authoritative websites and even contradicts with each other such that the LLM used for generation struggles in incorporating the retrieved information, *e.g.,* the first document just states they cannot be used together but separately without further context, the second document states they can be used for treating dry eye syndrome or ocular inflammation while the third document states the order of usage, however, various discussions held on this topic do not result in a definitive conclusion which finally hinders its effectiveness for question answering. Not to mention that for some complex queries that contains multiple factors, the top retrieved documents may only contains facts focusing on one factor and ignores documents with respect to other factors, which would severely hinder the performance.

Given the above limitations in current approaches, instead of solely relying on the relevance of documents for generation, we aim to integrate polyviews (*i.e.,* multiple views *w.r.t.* retrieval such as utility, complement, authoritativeness, timeliness and composibility) into consideration so as to promote its application in medical applications. However, the solution is quite non-trivial, which needs to tackle the following challenges: (**C1**) With multiple views to evaluate, how to measure them and its feasibility in real-world applications remains unknown. (**C2**) With the evaluated results of multiple views, in real-world applications what we needed is actually an integrated scoring strategy that comprehensively evaluates each view, how to develop a reasonable and applicable ranking strategy to combine the precedent views remains unanswered. (**C3**) The lack of benchmark data that evaluates the retrieval performance of a model from multiple views prohibits us from further developing our model.

Multi-view scoring slows down real time QnA.

To this end, we propose PolyRag. In particular, given that there are many available small but performant models, we carefully allocate storage to make this modeling feasible. (**C1**) To comprehensively integrate the results of each view, we transform the modeling of ranking strategy to a multi-reward problem and find the mixture of different views. (**C2**) Due to the scarcity of real-world benchmarks for evaluation, to bridge the gap we propose PolyEval, which is a benchmark consists of queries and documents collected from real-world healthcare scenarios (including medical policy, hospital recommendation and medical care) with multiple tagging (*e.g.,* timeliness, authoritativeness) on them (**C3**). With the polyviews gained from the precedent procedures, we apply the retrieved top-k documents and call an LLM for knowledge-augmented generation. We evaluate the proposed PolyRag on multiple tasks and extensive experiments and analysis on PolyEval have demonstrated the superiority of the proposed PolyRag.

2 Related Work

Retrieval-augmented generation (RAG) approaches which empower large language models (LLMs) with additional knowledge and henceforth less need for additional training (Gao et al., 2023b; Fan et al., 2024; Gupta et al., 2024; Nguyen et al., 2024) have been successfully applied to various fields(Sun et al., 2023; Zhang et al., 2024b; Shi et al., 2024b; Golatkar et al., 2024; Zhao et al., 2024) including recommender systems(Contal and McGoldrick, 2024; Rao and Lin, 2024; Zeng et al., 2024), question answering(Asai et al., 2024; Wang et al., 2025) and so on. Among them, question answering in medical applications poses significant challenges due to their high professionalism and low fault-tolerance characteristics. Existing approaches for medical-based RAG have been studying additional knowledge acquisition(Jin et al., 2023; Wang et al., 2024), query construction(Chen et al., 2025; Sohn et al., 2024), complex retrieval strategy(Wu et al., 2024; Xiong et al., 2024; Tang et al., 2025), complex reasoning(Verma et al., 2025; Li et al., 2024; Zafar et al., 2025) and so on with focus on better retrieval strategy from external source and better utilization strategy when employ LLMs for answer generation.

Open issues. Few research works consider multiple perspectives of the retrieval results and in this work we delve into a direction that can be directly integrated into these existing pipelines where we investigate on how to incorporate retrieval from polyviews for downstream tasks and henceforth promoting retrieval.

Let's say we have 3 documents: d1, d2, d3, and a query q.

- dj : the j-th document (so j = 1 for d1, 2 for d2...)
- Vi,j : score of d_j under view i
- wi : weight of view i (how important it is)

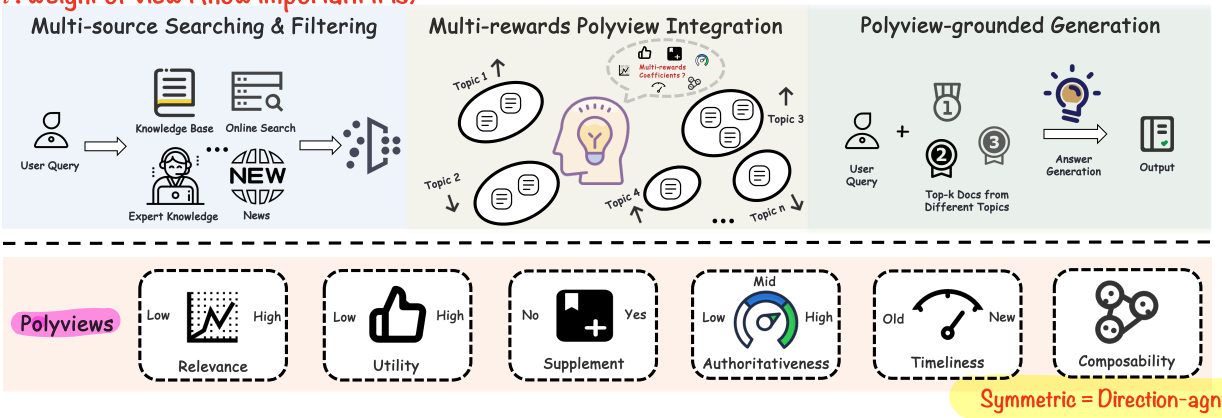


Figure 2: The proposed POLYRAG framework.

Symmetric = Direction-agnostic
→ Means R(q, d) = R(d, q)
→ Swapping query and doc doesn't change the relevance score.

3 The Proposed Approach

3.1 Overview

The task of retrieving top critical documents from previous searching and filtering stage is equivalent to comprehensively evaluate the input documents, *i.e.,* evaluate the retrieved document from m polyviews $\mathcal{V}$. For simplicity, with the assumption that multiple polyviews are independent, given an input query q, a document d ($d \in \mathcal{D} = \{d_1, d_2, ..., d_n\}$), where we first evaluate each document independently as follows:

Document pool

Equation for final score for each doc using multiple views

$$\mathbf{P}(d_j \mid \mathcal{V}_{1,j}, \dots, \mathcal{V}_{m,j}) = \prod_{i=1}^{m} (\mathbf{P}(d_j \mid \mathcal{V}_{i,j}))^{w_i}, \tag{1}$$

where $\mathcal{V}_{i,j}$, w_i denote the jth document evaluate from the ith view regarding the input query q, the weight of ith view respectively. Given some predefined constraints $\mathbb{C}$, we can obtain top-ranking documents $\mathcal{D}_{\text{Top}}$:

$$\mathcal{D}_{\text{Top}} = \{d \in \mathcal{D} \quad s.t. \quad \mathbb{C}\} \tag{2}$$

In this work, we propose POLYRAG, as shown in Figure 2. With the multi-source searching and filtering results, POLYRAG firstly embrace varied views for evaluation of each retrieved document (detailed in Section 3.2) and further pursuing integrated polyviews via a multi-rewards based view-mixture mechanism (detailed in Section 3.3), then incorporating the derived polyview-grounded knowledge for answer generation (detailed in Section 3.4).

3.2 Through Different Lenses: A Document Evaluated via Polyviews

In this paper, we pre-define 6 polyviews, *i.e.,* *Relevance* ($\mathcal{R}$), *Utility* ($\mathcal{U}$), *Supplement* ($\mathcal{S}$), *Authoritativeness* ($\mathcal{A}$), *Timeliness* ($\mathcal{T}$), *Composibility* ($\mathcal{C}$, which is used as a retrieval constraint) and detail the estimation of each in the following.

Relevance View is a case of symmetric retrieval, which is designed to be direction-agnostic. With an off-the-shelf model $\mathbf{E}$ (which could be a dense retriever followed by a predefined metric $\mathcal{M}$ such as cosine similarity for simplicity or large language models by designing instruction $\mathbb{INS}_{\mathcal{R}}$), we can efficiently obtain the *Relevance* score between the query and document as follows:

Embedding

Prompt

$$\mathcal{R}(q, d) = \begin{cases} \mathbf{P}_{LLM}(d|q, \mathbb{INS}_{\mathcal{R}}), & \text{with LLM;} \\ \mathcal{M}(\mathbf{E}(q), \mathbf{E}(d)), & \text{otherwise.} \end{cases} \tag{3}$$

Probability that the LLM thinks d is relevant to query q, based on INSR.

This is retrieval process of traditional RAG

However, *Relevance* cannot guarantee usefulness, where we introduce asymmetric retrieval *i.e.,* *Utility* View that measures the extent that one document is useful for assisting an LLM to answer the given query, which is modelled by the probability of generating correct answer a with a specific LLM, by designing an appropriate instruction $\mathbb{INS}_{\mathcal{U}}$ to guide the LLM, we can calculate the *Utility* of a document *w.r.t.* the input query as follows:

$$\mathcal{U}(d|q, a) = \mathbf{P}_{LLM}(a|q, d, \mathbb{INS}_{\mathcal{U}}). \tag{4}$$

Oftentimes there are documents that do not directly answer the query but they can provide additional knowledge, background information, or alternatives that help users to make more informed decisions or better understand the treatment process, where we define it as the *Supplement* View of a document *w.r.t.* the input query, with a carefully designed $\mathbb{INS}_{\mathcal{S}}$ to guide the LLM for estimating *Supplement*, we can formalize it as follows:

$$\mathbf{S}(d|q) = \mathbf{P}_{LLM}(d|q, \mathbb{INS}_{\mathcal{S}}). \tag{5}$$

Besides, given the retrieved documents from previous stage, it is of great significance to take

R: For each doc di , compute how similar it is to query q using either:

- Cosine similarity between embeddings: R(q, d$_i$) = M(E(q), E(d$_i$)), or
- LLM scoring with prompt INSR: R(q, d$_i$) = P$_{llm}$(d$_i$ | q, INSR)

U: For each doc di, compute how well it helps the LLM generate the correct answer a for query q.

S: For each doc di, assess if it provides helpful background info (even if not the direct answer).

Compositility View ensures topic diversity across top docs.

- Each doc di is assigned a topic cluster Ck.
- Done via LLM or clustering (e.g., KMeans) by maximizing Cd.
- Assign to the topic where it most likely belongs.

into account the *Authoritativeness* and *Timeliness* Views of them, since that for scenarios with strong professionalism, *i.e.,* medical applications in our case, medical treatments recommended by different sources, such as professional doctors and individual accounts, can vary greatly. Additionally, medical policies and practices may evolve over time. Therefore, keeping track of these two dimensions is crucial and here we denote these two dimensions of document d as $\mathcal{A}(d)$ and $\mathcal{T}(d)$[2]. Moreover, the retrieved documents might cover multiple topics *w.r.t.* the input query and directly ranking may lead to top documents focusing on partial topics, therefore, we introduce *Composibility* View to account for the difference of topics among them, where the topic of each document can be assigned via an LLM or clustering algorithms to maximize its assigning probability as follows:

$$\boxed{\mathcal{C}_d = \arg\max_k \mathbf{P}(C_k|d_i) \approx \arg\max_k \mathbf{P}(d_i|C_k)\mathbf{P}(C_k)} \tag{6}$$

3.3 A Cord of Three Strands is Not Quickly Broken: Multi-rewards Boosted Polyview Integration

Given the polyview evaluation results, to efficiently incorporate them for downstream generation, motivated by the idea and marvelous performance in simple rewards-driven reinforcement learning, here we model the integration as multi-rewards integration to obtain an effective mixture of polyviews, for each document d from $\mathcal{D}$, the polyview integration score can be formalized as follows:

Weights assigned to views

$$\boxed{y_d = \alpha_1 d_{\mathcal{R}} + \alpha_2 d_{\mathcal{U}} + \alpha_3 d_{\mathcal{S}} + \alpha_4 d_{\mathcal{A}} + \alpha_5 d_{\mathcal{T}},} \tag{7}$$

*

Does task-specific weight tuning (α1–α5) require manual tuning per domain, or can it be learned dynamically?

where the coefficients can be obtained either by expertise designation or learning from models. With the polyview integrated score, we can obtain the top-ranking documents $\mathcal{D}_{\text{Top}}$ under the *Composibility* constraints so that top-ranking documents can cover different topics *w.r.t.* the input query:

$$\left\|\mathcal{C}_d, d \in \mathcal{D}_{\text{Top}}\right\| \approx \left\|\mathcal{C}_d, d \in \mathcal{D}\right\|. \tag{8}$$

3.4 Polyview-grounded Generation

With the input query q and the polyview-grounded knowledge $\mathcal{P}$ that scatter across different topics related to the query, we can directly call an LLM (it can also be fine-tuned in a supervised manner), where its knowledge-augmented generation output o can be formalized as follows:

*

What happens when A(d) and T(d) (timeliness) conflict?
E.g., a newer blog post contradicts an older but authoritative guideline.

Does the integration mechanism (Eq. 7) adapt to prioritize recency over authority in evolving fields like COVID treatments?

$$o^* = \arg\max_o \mathbf{P}(o|q, \mathcal{P}), \tag{9}$$

where $\mathbf{P}(o|q,\mathcal{P})$ is the probability of the output o given the query q and the external documents $\mathcal{P}$, and $\arg\max$ denotes the argument of the maximum, i.e., the answer o for which $\mathbf{P}(o|q,\mathcal{P})$ is maximized.

4 Benchmark

We will first describe the characteristics of POLYEVAL and then delve into its creation process.

4.1 Characteristics

To ensure that POLYEVAL can be representatives of real-world medical application user cases, we carefully design it to be diverse in the following three perspectives.

- **Domain Type**: POLYEVAL contains questions from diverse domains including Medical Policy, Healthcare, Hospital & Doctor Inquiry in order to cover different real-world medical scenarios.
- **Query Intent**: Given questions in each domain, they encompass various types of real user intents, *e.g.,* Medical Insurance Balance in Medical Policy domain, Medication Inquiry in Healthcare domain in order to comprehensively represent user needs.
- **Annotation Dimension**: Given a query, for each retrieved document, it is annotated with tags on *relevance*, *complement*, *utility*, *publish date* and *authority level*.

4.2 Benchmark Creation

4.2.1 Data Collection

We collect 1,447 real-world user queries from a large-scale online platform that offers medical-related services in China, where its distribution of domain type and query intent is illustrated in Figure 3[3]. Given each query, we perform multi-source (including expert knowledge, online search engine, knowledge bases and news) documents searching to find relevant documents for annotation. In sum, we have collected 21,276 documents, making 14.7 documents for each query on average.

[2] We approximate $\mathcal{A}(d)$ via $\mathcal{A}(d_{source})$ for simplicity to reduce tagging costs, where the $\mathcal{A}(d_{source})$ is annotated by human annotators. For $\mathcal{T}(d)$, we employ efficient tool for date extraction.

[3] Due to space limit, we only used the first-level categories when drawing the query intent distribution. In total, there are 40 labels when considering the second-level categories.

*

They use a weighted linear combination of view scores:

- This is inspired by RL-style reward aggregation, but not RL.
- The weights αi are manually set or learned,
- No policy learning, no reward optimization step.

Example, Query q: Is BrandX safe for diabetic patients?
Retrieved Doc: A health article mentioning BrandX.
Relevance: Talks directly about BrandX and diabetes, Utility: Helpful or not (e.g., includes dosage info).
Supplement: Gives background on how BrandX works, Timeliness: Published in 2017 (may be outdated), Authority: From a government health site.

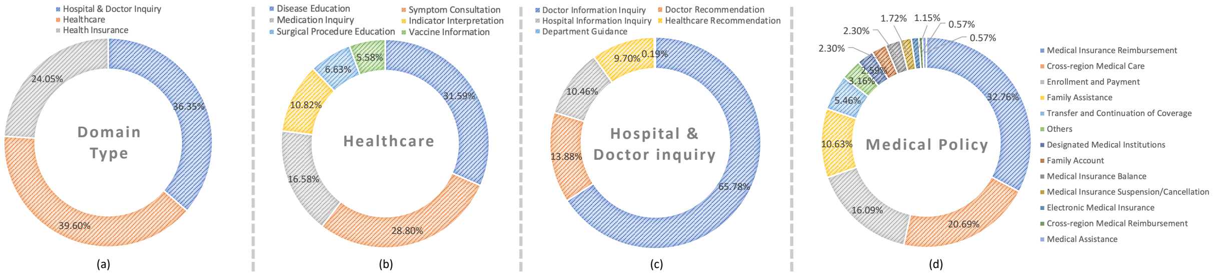


Figure 3: Data distribution of PolyEval, where Figure (a) denotes the domain type distribution and Figure (b-d) denote the query intent distribution within each domain.

4.2.2 Annotation Details

Overall, PolyEval is annotated by human annotators or automated tools. For each query and its associated documents, three highly-skilled annotators who have received professional medical training are involved for document *relevance*, *complement* and *utility* annotation and the annotation result is "accepted" if at least two annotators reach an agreement unless it is "rejected". For *authority level* of document, we approximate it via the *authority level* of its source, *i.e.*, we firstly collect abundant information from multiple sources such as medical-related websites and random sample information from them, and then ask human annotators to judge the overall authority of these sources and finally come up with the *authority level*. For *publish date* of document, we employ efficient automated tools for date extraction.

5 Experiments

5.1 Experimental Setup

5.1.1 Tasks

We evaluate our proposed PolyRag and multiple baselines for retrieval and generation on PolyEval and evaluate the performance of retrieval via metrics HIT, NDCG, and generation via judge model (*e.g.,* GPT 4). To better demonstrate the difference between domains in PolyRag, we denote data of domain Healthcare, Hospital & Doctor Inquiry, Medical Policy as CARE, INQUIRY and POLICY respectively for simplicity.

5.1.2 Baselines

We evaluate models augmented with retrieval via publicly available retrieval model including BM25, GTE (Li et al., 2023), BGE-M3 (Chen et al., 2024), jina embedding v3 (Sturua et al., 2024). With the top-k retrieved documents, we directly call strong publicly available pre-trained LLMs, $\text{Qwen2.5}_{7B,14B,32B}$ (Yang et al., 2024) for generation.

5.1.3 Training, Generation and Evaluation Details.

Our training data includes randomly sampled <query,document,label> triples (which are excluded from PolyEval)[4] from a large-scale medical service platform in China to train our model for evaluating polyviews. All experiments are conducted using 4 NVIDIA A100 GPUs. For *Relevance* and *Supplement* evaluation, we utilize open-source Llama Factory[5] to finetune small-scale $\text{Qwen2.5}_{1.5B}$ and adopt Lora tuning for 5 epoch with a learning rate of 5e-5, a batch size of 4 and a cosine learning rate scheduler. As for *Utility* evaluation, we incorporate BGE-M3 owing to its superior performance in a variety of benchmark leaderboards and distill the marvelous power of LLM in evaluating utility into it, where $\mathcal{M}(\cdot)$ is defined as cosine similarity. We train the utility model for 5 epochs with a learning rate of 1e-5, a batch size of 16 for each device, a warm-up ratio of 0.2, the passage window size of 50 and the temperature parameter τ set to 0.05 following (Gan et al., 2024). For *Composibility* evaluation, we borrow the embedding from *Utility* and conduct clustering via DBSCAN (Ester et al., 1996). For all generation tasks, we utilize vLLM (Kwon et al., 2023) for inference speed-up and set the temperature to 0 for reproducibility and max token parameter to 1. We set $[\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5]$ is set to [0.35, 0.35, 0.1, 0.1, 0.1] for INQUIRY and POLICY and [0.35, 0.35, 0.1, 0.2, 0.0] for CARE for simplicity. For generation evaluation, we directly call private commercial LLM GPT4 to conduct answer statement generation and the judgement (*i.e.,* circumstances

[4] For *Relevance* and *Supplement* evaluation, the label is binary, *i.e.,* 0 or 1 while for *Utility* evaluation the label is a float number generated by a powerful LLM.

[5] https://github.com/hiyouga/LLaMA-Factory

How does POLYRAG compare if only 1 or 2 views are used? Is all 5 necessary for gains?

Table 1: Overall retrieval performance (%) evaluation on PolyEval, here k is set to 3 for simplicity.

Retrieval	CARE		INQUIRY		POLICY	
	HIT	NDCG	HIT	NDCG	HIT	NDCG
BM25	26.6	26.6	22.3	22.7	28.2	28.6
GTE	38.7	39.4	31.1	31.7	31.4	32.0
BGE-M3	40.0	40.8	34.8	35.7	33.7	34.6
jina	42.8	43.7	33.5	34.7	35.9	36.9
PolyRag	**47.1**	**48.3**	**38.1**	**39.1**	**42.8**	**44.5**

Table 2: Generation performance (%) evaluation on CARE using Top-3 Documents for Retrieval.

Retrieval	Generation	$\mathbb{R}_c \uparrow$	$\mathbb{R}_i \downarrow$	$\mathbb{R}_n$	$\mathbb{N}_c \uparrow$	$\mathbb{N}_i \downarrow$	$\mathbb{N}_n$
BM25	Qwen2.5_{7B}	35.7	8.23	53.4	3.02	0.62	4.72
	Qwen2.5_{14B}	54.6	6.97	35.3	4.39	0.55	3.01
	Qwen2.5_{32B}	57.5	6.31	33.2	4.62	0.50	2.80
GTE	Qwen2.5_{7B}	34.5	7.70	55.2	3.16	0.72	5.03
	Qwen2.5_{14B}	53.3	7.02	36.3	4.56	0.60	3.26
	Qwen2.5_{32B}	55.3	6.31	35.3	4.65	0.52	3.09
BGE-M3	Qwen2.5_{7B}	36.2	8.55	51.3	3.25	0.72	5.06
	Qwen2.5_{14B}	54.8	6.93	34.9	4.48	0.56	2.97
	Qwen2.5_{32B}	57.3	6.89	32.6	4.75	0.56	2.83
jina	Qwen2.5_{7B}	38.4	**6.96**	51.5	3.50	0.62	5.02
	Qwen2.5_{14B}	55.5	6.67	34.6	4.53	0.54	3.01
	Qwen2.5_{32B}	57.0	6.87	33.0	4.65	0.57	2.84
PolyRag	Qwen2.5_{7B}	**60.9**	7.65	27.5	**4.71**	**0.52**	2.23
	Qwen2.5_{14B}	**69.2**	**4.80**	22.0	**5.32**	**0.36**	1.72
	Qwen2.5_{32B}	**71.6**	**5.40**	20.5	**5.39**	**0.38**	1.53

correct, incorrect and not mentioned) between answer statement and ground truth and $\mathbb{N}_c$, $\mathbb{R}_c$ denote the count and ratio of the given circumstance c. Finally, we have listed all prompt templates in the Appendix.

5.2 Results and Analysis

5.2.1 Main Results

From the empirical results on retrieval and generation tasks (Table 1 and Table 2), we can summarize the major findings as follows:

- **PolyRag largely improves the performance of retrieval and generation for knowledge-intensive tasks.** We only list the retrieval results due to the fact that refusal rate is high without retrieval (*e.g.*, for INQUIRY the refusal rate is as high as 59.7% for Qwen2.5_{7B})). By comprehensively combining retrieval and generation metrics defining the correct count, correct ratio, incorrect count, incorrect ratio, we can find that PolyRag consistently performs well in different tasks and metrics.

- **Both time-evolving and authoritative-sensitive tasks benefit more from PolyRag.** A large margin of improvement can be found in POLICY as it is more sensitive to timeliness and authoritativeness compared to task such as CARE, which depends more on the authoritativeness since the improvement of the treatment takes a lot of time.

- **More customization of PolyRag *w.r.t.* downstream tasks deserves more attention.** We take a trivial step to assign weights to tasks in PolyEval, however, the ablation study demonstrates the importance of different views varies across different tasks, hence more attention should be devoted to its customization since each task comes with areas of emphasis.

5.2.2 Feasibility Analysis and Broader Impact

For industrial platform that directly serves user queries, low-latency inference is of great significance. In PolyRag, we utilize polyviews for a more comprehensive way of information integration that incorporate multiple models in this progress, where the overall procedure is illustrated in the upper part of Figure 2. By flexibly incorporating multiple small-scale models and the concurrency and GPU Segmentation mechanisms, the polyview-based integration stage can be deployed using a L20 GPU with latency around 200ms given an user query with an average of 15 documents where the total length exceeds 8k tokens. Besides medical applications, for the broader application, the idea of PolyRag can also be applied to other domains such as finance where the authoritativeness and timeliness of information greatly matters.

Paper claims real-world use, but lacks benchmarks comparing individual models or view-only baselines vs.

6 Conclusion and Future Work

In this work, we propose PolyRag that incorporates varied views for evaluation of each retrieved document and then pursues integrated polyviews via a multi-reward based view-mixture mechanism, which finally incorporates the derived polyview-grounded knowledge for answer generation. To bridge the evaluation gap we also propose PolyEval, a benchmark consists of queries and documents collected from real-world medical scenarios with multiple annotation on them. Experiments and analysis on PolyEval have demonstrated the superiority of PolyRag. Nevertheless, we take a trivial step for the multi-rewards mixture and more complicated approaches requires further research. In the future, we would like to explore multi-modal retrieval integration and apply the proposed PolyRag to other scenarios such as finance.

R: A–E grade, via prompt scoring.
U: Float score. Generated using LLMs or BGE-M3.
S: 0 or 1. Human-labeled or inferred with LLM.
A: Categorical or scored. Based on the document source (e.g., gov site = high).
T: Derived date → becomes recency score

Gaps

1. Does not isolate the impact of each individual view.
2. No comparison against other advanced multi-step/multi-signal RAGs. Such as MedRAG, BioRAG, DRAGIN etc.
3. Missing: Trade-off analysis of accuracy vs latency
4. No mechanism to resolve conflicting signals (newer but less authoritative source).