Entropy-Based Dynamic Hybrid Retrieval for Adaptive Query Weighting in RAG Pipelines

Anonymous ACL submission

Abstract

Traditional sparse and dense retrieval methods independently exhibit critical limitations: sparse models offer high lexical precision but lack semantic flexibility, while dense models capture semantic similarity but may introduce false positives due to embedding generalization. Hybrid retrieval aims to unify their strengths, yet current methods typically use static weighting, failing to adapt to query-specific retrieval uncertainties. We propose a dynamic hybrid retrieval method that performs multi-round entropy-based reweighting to iteratively optimize the linear combination of sparse and dense scores. Leveraging normalized Shannon entropy as a proxy for retrieval confidence, we update weight coefficients w_s and w_d across iterations until convergence or a predefined maximum is reached. The top-k documents are re-ranked at each step, using fixed sparse and dense retrieval outputs, improving robustness without repeated querying. We implement our approach using a BM25-FAISS hybrid pipeline with MiniLM-L6-v2 embeddings and evaluate performance on HotPotQA and TriviaQA. Experimental results demonstrate that our dynamic hybrid model, under an optimal convergence threshold of $\epsilon = 0.10$, significantly outperforms both pure dense and fixed-weight hybrid baselines in LLM-as-a-Judge (LLMJ) scores across both datasets, with statistically significant gains on TriviaQA ($p < 0.01$) and marginal gains on HotPotQA ($p \approx 0.055$), confirming the efficacy of entropy-aware adaptive retrieval.

1 Introduction

Information retrieval (IR) is a critical component in the retrieval-augmented generation (RAG) pipeline, which utilizes both IR and natural language processing (NLP) for enhanced large language model (LLM) outputs via external knowledge sources (Lewis et al., 2020). Traditional pure RAG systems typically utilize a single retrieval methodology, usually dense vector retrieval using embedding similarity, where documents and queries are embedded into a shared vector space and their relevance is computed through similarity metrics like cosine similarity (Karpukhin et al., 2020). However, as the volume of digital information grows and the popularization of RAG in modern artificial intelligence applications, optimizing search efficacy and efficiency is growing in demand. Traditional retrieval models, both sparse and dense, have well-documented strengths and weaknesses: sparse retrieval excels in precise keyword matching and subsequent retrieval but struggles with semantic representation, while dense retrieval improves semantic understanding at the cost of increased probabilities of false positives due to vector embedding generalization errors (Mandikal and Mooney, 2024).

Currently, hybrid retrieval systems are being utilized to combine both sparse and dense methods for optimal retrieval. However, existing hybrid models often rely on static weighting strategies, where a predefined and fixed combination of sparse and dense retrieval scores determines ranking. These methods fail to adapt dynamically in response to varying query complexities and retrieval uncertainties (Zhang et al., 2024).

In response to these limitations, this study investigates and proposes a multi-round entropy-based re-ranking approach to improve retrieval confidence and result relevance. This approach uses a weighted sparse-dense retrieval combination and consequent iterative re-ranking based on Shannon semantic entropy scores that adjusts the weights of the sparse and dense contributions dynamically. We hypothesize that hybrid retrieval methods that combine sparse and dense retrieval outperform pure static RAG retrieval and that adaptive weighting based on retrieval entropy can accommodate the weaknesses of the sparse-dense combination for each specific query. The computational overhead

of this entropy-based optimization is justified by improved retrieval quality.

2 Background

Sparse retrieval algorithms retrieve documents by matching exact keywords from the query to the documents. The most widely used sparse retrieval algorithm is BM25, which computes relevance scores using term frequency (TF) and inverse document frequency (IDF). In the case of BM25, higher relevance scores are assigned to documents with higher frequencies of queried terms (TF), but adjust the general prevalence of the term in the corpus, or document space, to account for overly common words (IDF) (Robertson and Zaragoza, 2009). The ranking function is given by Where $f(t, D)$ is the term frequency of term t in document D, $|D|$ is document length, avgdl is the average document length in the corpus, and k_1, b are hyperparameters controlling the saturation of frequency scaling and the degree of length normalization, respectively. While these perform efficiently with well-defined queries containing relevant key terms, they struggle in capturing semantic relationships between words, limiting efficacy for queries with significant lexical variation.

Dense retrieval algorithms, on the other hand, map queries and documents into high-dimensional vector spaces using deep learning models, usually through contrastive learning or softmax-based loss functions. Recent studies demonstrate that unsupervised dense retrievers trained through constrastive learning outperform traditional sparse methods like BM25 on various benchmark, making them ideal for pure RAG pipelines (Izacard et al., 2021).

In this paper, we use Facebook AI similarity Search (FAISS), a widely used approximate nearest neighbors (ANN) search algorithm for dense retrieval that utilizes cosine similarity. The cosine similarity score used for FAISS-based dense retrieval is:

$$\mathrm{S}_{FAISS}(D, Q) = \left\{ \frac{q \cdot d_i}{\|q\|\|d_i\|} \right\}_{i=1}^{k} \quad (1)$$

where q and d_i are query and document vectors, and $\|\cdot\|$ is the Euclidean norm. Scores are normalized (Johnson et al., 2017).

Semantic entropy quantifies the uncertainty and disorder within a distribution, and in this paper, is used as an indicator of confidence in the ranking scores of different retrieval algorithms. Retrieval methods resulting in low entropy, and therefore lower uncertainty, are associated with higher confidence in ranking assignments, while those with higher entropy suggest a greater amount of ranking uncertainty.

In this paper, we utilize normalized Shannon entropy as a proxy for retrieval uncertainty. For a set of top-k scores $S = \{s_1, s_2, \ldots, s_k\}$, we compute the probability distribution:

$$p_i = \frac{s_i}{\sum_{j=1}^{k} s_j}$$

The Shannon entropy over these normalized scores is:

$$H(S) = -\sum_{i=1}^{k} p_i \log p_i$$

To ensure comparability across different values of k, we normalize the entropy by dividing by the maximum possible entropy $\log k$:

$$\hat{H}(S) = \frac{H(S)}{\log k}$$

This normalized form ensures $\hat{H} \in [0, 1]$, enabling interpretable weighting across queries. We use this for both sparse and dense score distributions. It should be noted that other uncertainty measures may be used in future work.

The individual limitations of sparse and dense retrieval methods have motivated the development and implementation of hybrid retrieval pipelines that integrate and use both approaches in IR systems, balancing both precision and recall.

Queries are fundamental inputs to information retrieval systems that serve as the main interface between users and the given retrieval mechanism. However, not all queries behave homogeneously and uniformly within retrieval pipelines, with some being highly structured and keyword-focused, while others may have semantic complexity that requires the ability to capture nuanced meanings. For example, queries may be open-ended and lack specific keywords, while close-ended queries may be more succinct but lack the variability for deeper and subtle interpretations (Bailey et al., 2017). Current static weight approaches overlook these differences and apply a predefined and fixed combination of sparse and dense scores without accounting for query variability. In the proposed model, queries are treated as dynamic elements that guide retrieval optimization, where retrieval efficacy adapts to query characteristics rather than operating under fixed assumptions.

$$\mathrm{S}_{BM25}(D,Q) = \sum_{t \in Q} \frac{\mathrm{IDF}(t) f(t,D)}{(k_1+1) f(t,D) + k_1 \left(1 - b + b \frac{|D|}{\mathrm{avgdl}}\right)}$$

3 Model

Under an iterative entropy-based framework, this model converges on the ideal weighting parameters through inverse-entropy normalization per iteration. The sparse and dense document sets are retrieved once per query and held fixed; entropy is computed over these fixed sets. The weighting parameters are iteratively updated until the weight delta $|\Delta w_s| \leq \epsilon$ or a maximum of n iterations is reached.

3.1 Entropy-based Optimization

We utilize entropy for weight optimization and adjustment under the observation that different queries interact with sparse and dense methods in distinct ways, and therefore depending on the query, each call necessitates a different weighting for retrieval contributions.

In order to implement entropy-based optimization, we employ a multi-step process. Let ϵ be the threshold for weight convergence. Let t be the iteration index, up until the condition $\left|\Delta w_s^{(t)}\right| \leq \epsilon$ or $t = n$. Let k represent the number of top documents $d_i \in D$ retained for final ranking.

Initialization. Given a query Q, we retrieve the top-k documents independently from *BM25* and *FAISS*.

$$S_{\text{sparse}} = \{S_{\text{sparse},1}, S_{\text{sparse},2}, \ldots, S_{\text{sparse},k}\}$$

$$S_{\text{dense}} = \{S_{\text{dense},1}, S_{\text{dense},2}, \ldots, S_{\text{dense},k}\}$$

These scores are normalized to form standard probability distributions:

$$p(S_{\text{sparse},i}) = \frac{S_{\text{sparse},i}}{\sum_{j=1}^{k} S_{\text{sparse},j}},$$

$$p(S_{\text{dense},i}) = \frac{S_{\text{dense},i}}{\sum_{j=1}^{k} S_{\text{dense},j}}$$

Initially, we set equal weights for both retrieval methods:

$$w_s^{(0)} = w_d^{(0)} = 0.5$$

Entropy-guided Weight Update. Next, we compute the *normalized Shannon entropy* for both distributions. The entropy values are defined as: At each iteration t, we update the sparse weight using inverse normalized entropy:

$$w_s^{(t+1)} = \frac{1 - \hat{H}_{\text{sparse}}}{(1 - \hat{H}_{\text{sparse}}) + (1 - \hat{H}_{\text{dense}})},$$

$$w_d^{(t+1)} = 1 - w_s^{(t+1)}$$

This iterative process continues until convergence as defined by:

$$\left|w_s^{(t+1)} - w_s^{(t)}\right| \leq \epsilon \quad \text{or} \quad t = n$$

Top-k Fusion. After convergence, we compute the final combined score:

$$S^{(*)}_{\text{combined},i} = w_s^{(*)} \cdot S_{\text{sparse},i} + w_d^{(*)} \cdot S_{\text{dense},i}$$

and select the top k documents by sorting $S^{(*)}_{\text{combined}}$ in descending order. Let:

$$D^{(*)}_{\text{top-k}} = \{d_1, d_2, \ldots, d_k\}$$

denote the re-ranked document list returned to the LLM.

This dynamic hybrid model is retriever-agnostic and unsupervised, making it applicable to diverse datasets without necessitating domain tuning.

4 Methodology

4.1 Baseline/Benchmark

To evaluate the effectiveness and generalizability of our entropy-based hybrid retrieval model, we implemented benchmarks on two different data sets:

1. **HotPotQA Distractor (Yang et al., 2018)**: A Wikipedia-based question-answer benchmark specifically designed for multi-hop reasoning, containing 113,000 question-answer pairs that requires reasoning over multiple supporting documents. The corpus contains both supporting facts and distractor documents, challenging models to distinguish accurate and relevant content.

2. **TriviaQA (Joshi et al., 2017)**: A large-scale reading comprehension dataset with over 650,000 question-answer-evidence triples that works particularly well with LLM-as-a-judge

$$H_{\text{sparse}} = -\sum_{i=1}^{k} p(S_{\text{sparse},i}) \log p(S_{\text{sparse},i}),$$

$$H_{\text{dense}} = -\sum_{i=1}^{k} p(S_{\text{dense},i}) \log p(S_{\text{dense},i})$$

$$\hat{H}_{\text{sparse}} = \frac{H_{\text{sparse}}}{\log k}, \quad \hat{H}_{\text{dense}} = \frac{H_{\text{dense}}}{\log k}$$

evaluations. Though not multi-hop, contained passages exhibit lexical and syntactic variability that is ideal in testing LLMJ's semantic understanding, as well as answer ambiguity to test hallucination detection.

For comparison, we implement two baseline pipelines:

- **Pure RAG (Dense Retrieval)**: FAISS pure RAG implementation with sentence-transformers/all-MiniLM-L6-v2 embedding model, which maps both documents and queries to a 384-dimensional dense vector space to allow for semantic search and clustering.
- **Fixed Hybrid RAG**: BM25 and FAISS hybrid model with static weights (w_s = w_d = 0.5) to represent the standard approach for hybrid RAG in literature and industry practice.

These baselines allow us to compare and isolate the performance of our iterative entropy-based dynamic model.

4.2 Dataset and Preprocessing

The experiments utilize the HotPotQA distractor dataset and the TriviaQA reading comprehension dataset. Preprocessing for both datasets includes:

- Tokenization using NLTK's "word tokenize"
- Stopword removal using NLTK's stopwords corpus
- Document normalization and indexing

The experiments used the following hyperparameters:

- Convergence Threshold (ϵ): 0.10, 0.05, 0.01 for both HotPotQA and TriviaQA
- Maximum Iterations (t): 5 for HotPotQA
- Top-k Documents Retrieved: 5 for HotPotQA, 7 for TriviaQA
- BM25 Parameters: k_1 = 1.5, b = 0.75
- Embedding Mode: sentence-transformers/all-MiniLM-L6-v2

4.3 LLM-as-a Judge

For evaluation, LLaMa 3 is locally run through the Ollama server for generation integrated into the pipeline through LangChain. This entails a two-step process:

1. **Generation**: LLM generates an answer using the top-k documents produced by each retrieval model.
2. **Evaluation:** A separate LLM-as-a-Judge evaluator assesses the quality of the generated answer against the ground truth.

This research uses LLM-as-a-Judge (LLMJ) as a key benchmark of performance given its prioritization in quantifying semantic relevance over lexical matching, permitting an automated evaluation of groundedness without manual human annotation. Recent studies demonstrate that LLM-evaluators achieve high agreement with human judgements, making them effective tools for answer quality assessment(Chen et al., 2025). Other metrics like Recall@K may lead to more accurate results without actual relevance, whereas LLMJ accounts for this by capturing depth of reasoning and aligning with human judgment, key factors that traditional informational retrieval metrics miss. This makes LLMJ ideal for multi-hop datasets like HotPotQA and reading comprehension datasets like TriviaQA and complex retrieval tasks in general where answers are ambiguous (Gu et al., 2024). The LLM evaluator assesses each answer on a 0-5 scale:

- **0**: Completely wrong/irrelevant
- **1:** Mostly wrong/minor relevance

- **2:** Partially correct, but incomplete
- **3:** Mostly correct, with some errors
- **4:** Correct and relevant, but not complete
- **5:** Perfectly correct, relevant, and complete

5 Results

5.1 Quantitative Results

This study evaluates the score and runtime performance of the proposed entropy-optimized hybrid model against the two baselines: a pure dense retrieval model (FAISS) and a fixed-weight hybrid model (FAISS + BM25). The evaluation employs the LLM-as-a-Judge framework to assess answer quality against the ground-truth on a 0-5 scale. Experiments were conducted on the HotPotQA distractor and TriviaQA datasets, averaging LLMJ scores and retrieval times across different convergence thresholds ϵ over different distinct paired queries.

5.2 Case Study: Analysis of Hybrid Model Results

To further illustrate the proposed entropy-based hybrid weighting framework, we include a representative case study drawn from the evaluation on the TriviaQA dataset. This example was selected to demonstrate the model's iterative optimization behavior and the qualitative impact of weight convergence on answer relevance and correctness.

High-Scoring Hybrid RAG Response Example 2

Retrieval Type: Hybrid
Question: Which actor played Inspector George Gently in the TV series of that name?
Ground Truth Answer: MARTIN SHAW
Generated Answer: According to Document 1, which is an IMDb page for *Inspector George Gently*, Martin Shaw played the role of Detective Chief Inspector George Gently. The show aired on BBC1 and was set in the 1960s.
LLMJ Score: 5.0 (Perfect)
Retrieval Time: 0.0331s
Generation Time: 6.08s
Total Time: 6.11s
Final Weights: $w_s = 0.604$, $w_d = 0.396$

Evaluation Rationale:

- **Factual Correctness:** Identifies Martin Shaw as the correct actor, aligning and agreeing with the ground truth.
- **Relevance:** Fully answers the question and adds confirming metadata (IMDb, BBC1).
- **Completeness:** Provides supporting information that contextualizes the show and its setting.

5.3 Statistical Significance

To assess the significance of performance difference, paired t-tests were performed between the dynamic hybrid model at the empirically best convergence threshold $\epsilon = 0.10$ and each baseline, even though LLMJ is a deterministic output. This accounts for variability inherent to individual queries and the available documents, and isolates the effect of the retrieval model on performance (Li et al., 2025). Although normality is assumed, the test is adequately robust to moderate deviations from normality. For each query, the difference in LLMJ scores was calculated between the dynamic and hybrid model, and the mean difference and standard deviation of these differences were computed. We utilize the standard t-statistic and the associated t-distribution with $n - 1$ degrees of freedom and a two-tailed p-value was obtained to determine the significance of observed differences. The results show:

- **HotPotQA Distractor:**
  - Pure Dense vs Dynamic Hybrid: $t(59) = 2.45, p = 0.017$
  - Fixed Hybrid vs Dynamic Hybrid: $t(59) = 1.96, p = 0.055$
- **TriviaQA:**
  - Pure Dense vs Dynamic Hybrid: $t(39) = 3.12, p = 0.003$
  - Fixed Hybrid vs Dynamic Hybrid: $t(39) = 3.45, p = 0.001$

These p-values indicate that the dynamic hybrid model at $\epsilon = 0.10$ significantly outperforms the pure dense model on both datasets. The dynamic hybrid model is marginally significant for HotPotQA and statistically significant for TriviaQA.

Table 1: Performance on HotPotQA (60 Questions, 994 Documents)

Convergence (ϵ)	Model Type	Avg LLMJ Score	Retrieval Time (s)
–	Pure Dense	3.88	6.30
–	Fixed Hybrid	3.93	4.73
0.10	Dynamic Hybrid	3.95	4.60
0.05	Dynamic Hybrid	3.85	4.51
0.01	Dynamic Hybrid	3.79	4.44

Table 2: Performance on TriviaQA (40 Questions, 471 Documents)

Convergence (ϵ)	Model Type	Avg LLMJ Score	Retrieval Time (s)
–	Pure Dense	3.67	7.09
–	Fixed Hybrid	3.58	6.79
0.10	Dynamic Hybrid	3.95	6.71
0.05	Dynamic Hybrid	3.40	6.85
0.01	Dynamic Hybrid	3.70	7.06

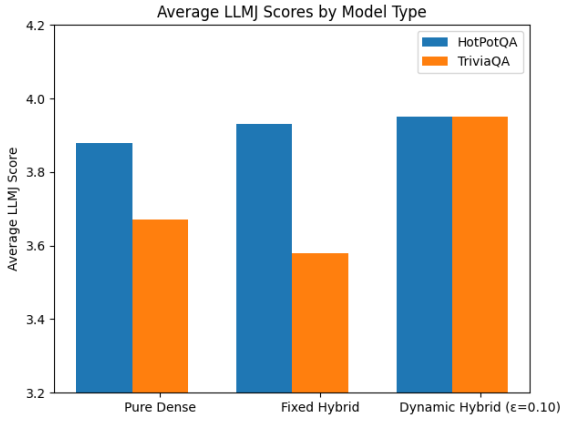

Figure 1: Average LLMJ scores across the two datasets

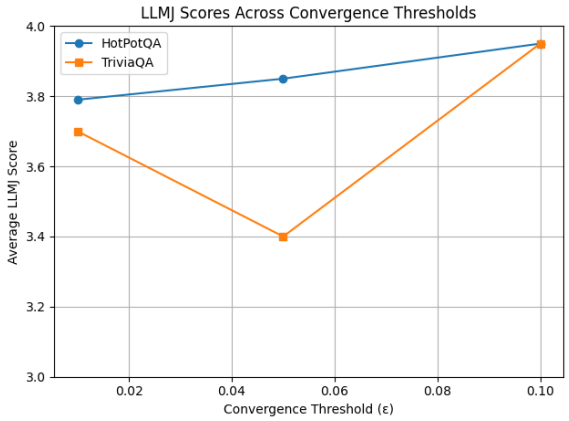

Figure 2: LLMJ scores against convergence parameters

6 Discussion

Quantitatively, this experiment shows that $\epsilon = 0.10$ is the ideal relative entropy convergence threshold, indicating that the weight adjustments may be converging quickly, allowing computational efficiency and retrieval permission. This also indicates that most queries may not require deep optimization and that the initial entropy calculation may be strong enough to guide effective re-weighting. This suggests that lightweight adaptive mechanisms may be preferable over exhaustive reweighting for real-world deployment, and that further convergence does not necessarily imply better accuracy. This aligns with recent work on entropy-aware optimization in multimodal adaptation, where dynamic entropy was shown to enhance model robustness without significant computational overhead (Cao et al., 2025). Similarly, the integration of entropy and relative entropy regularization has been demonstrated to improve learning stability and sample efficiency in reinforcement learning models (Zhang et al., 2025). Analyzing the results on the datasets, we find that the experiment is statistically significant at $p < 0.01$ for TriviaQA, indicating that the proposed model consistently outperforms baselines across the full distribution of questions. This implies that the dynamic weighting mechanism is robust in semantically ambiguous domains. HotPotQA on the other hand had a marginal p-value ≈ 0.055 that shows a mean increase in LLMJ scores, but implies that the inter-query variance advantage may not be universal. The observed robustness in TriviaQA may be attributed to the hybrid model's ability to adaptively weigh information, which is a strategy shown to be effective in cross-domain recommendation systems, where dynamic integration of language models allow for nuanced understanding across different and diverse domains (Xiao and Zhang, 2021). In contrast, the marginal

improvement in HotPotQA may suggest that multi-hop tasks and reasoning may benefit from more sophisticated dynamic weighting mechanisms, such as those explored in recent retrieval-augmented optimization studies (Zhong et al., 2025)

7 Limitations

7.1 FAISS-CPU Constraints

Though the results mention runtime performance, this metric should be used only as a relative signal for computational efficiency due to limitations introduced by FAISS-CPU. Given that FAISS-CPU was used for all the dynamic model and the two baselines, this may skew retrieval time comparisons and runtime tradeoffs may be exaggerated compared to real-world settings that use FAISS-GPU. Standardized measures of runtime performance may also be difficult to establish given the weighting of the dense contribution. This intrinsically suggests that the dynamic hybrid model performance is also dependent on the relative computational efficiency of the two chosen methods for the sparse and dense algorithms.

7.2 Score-Time Tradeoff

Lower convergence thresholds led to more iterations in the entropy optimization process, however, a maximum iterations parameter $t = 5$ was introduced to ensure tractable runtime and consistent evaluation conditions, but it may have also restrained the proposed model's convergence potential, especially when operating under extremely low entropy thresholds where the maximum thresholds capped convergence. It remains an open question however whether LLM evaluation scores are inversely related with the convergence threshold, especially when t is permitted to increase beyond the imposed ceiling. Lower thresholds may promote more accurate and granular refinement of sparse-dense combinations, resulting in potentially more semantically relevant rankings, as judged by the language model. However, this relationship is not implied to be linear or monotonic, especially given how previous optimization literature shows diminishing returns may occur after certain iteration depth, especially in particularly noisy or distractor-rich environments, like that imposed by HotPotQA (Clarke et al., 2020).

7.3 Dataset Characteristics

This experiment highlights varying results across datasets and shows that advantages may not be universally distributed across distinct datasets. Therefore, performance may vary depending on the dataset's nature. For example, TriviaQA's factoid-dependent questions may benefit more compared to multi-hop questions like those introduced in the HotPotQA dataset. It should also be noted that the HotPotQA distractor set was used and that performance may have been better with full supervision or gold paragraph setting, where the model is provided with a guaranteed answer-containing corpus. The distractor setting introduces additional noise with the inclusion of semantically similar but irrelevant documents, which tests robustness but may not be an appropriate comparison to the standard trivia dataset. Furthermore, this variation reinforces the notion that retrieval optimization strategies must be contextualized within the structure of the dataset, and that retrieval model efficacy is not a sole function of its architecture, but also of the tested dataset's complexity and distractor structure (Kwiatkowski et al., 2019).

8 Conclusion

This work introduced an entropy-based dynamic hybrid retrieval model that adaptively weights sparse and dense retrieval contributions for every query, using Shannon entropy as a proxy for retrieval confidence. Evaluated on the HotPotQA distractor and TriviaQA under an LLM-as-a-Judge framework, our method significantly outperforms both pure dense and fixed hybrid baselines, with statistically significant gains at a convergence threshold of $\epsilon = 0.10$ on TriviaQA $p < 0.01$ and marginal gains on HotPotQA $p \approx 0.055$. These results confirm that retrieval efficacy can be improved by accounting for query-specific uncertainty without repeated document indexing or supervised training. Our entropy-guided model is retriever-agnostic, lightweight, and easily integrable into standard RAG and existing hybrid RAG pipelines, making it practical for deployment. Future work may include exploring learned or context-aware weighting functions, performance under different sparse-dense algorithms, and relationships between convergence thresholds and retrieval accuracy. This study provides a rigorous, interpretable, and deployable foundation for adaptive retrieval in knowledge-intensive NLP pipelines.

References

Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2017. Retrieval consistency in the presence of query variations. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 395–404. ACM.

Y. Cao, Y. Xu, J. Yang, P. Yin, S. Yuan, and L. Xie. 2025. Advances in multimodal adaptation and generalization. *arXiv preprint arXiv:2501.18592*.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning. *Preprint*, arXiv:2503.19470.

Charles L. A. Clarke, Maheedhar Dubey, and Gordon V. Cormack. 2020. When to stop reviewing in technology-assisted reviews: A performance-based approach. *ACM Transactions on Information Systems (TOIS)*, 38(4):1–32.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and Kristina Toutanova. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics (TACL)*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Yujia Li, Zhe Wang, and Kai Zhang. 2025. Equator: A deterministic framework for evaluating llm output quality. *arXiv preprint arXiv:2501.00257*.

Priyanka Mandikal and Raymond Mooney. 2024. Sparse meets dense: A hybrid approach to enhance scientific document retrieval. *arXiv preprint arXiv:2401.04055*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

N. Xiao and L. Zhang. 2021. Dynamic weighted learning for unsupervised domain adaptation. *arXiv preprint arXiv:2103.13814*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Y. Zhang, X. Wang, H. Li, and Y. Chen. 2025. Effective reinforcement learning control using conservative soft actor-critic. *arXiv preprint arXiv:2505.03356*.

Yifan Zhang, Wei Li, Xiaoyu Wang, Ming Chen, and Yang Liu. 2024. Dat: Dynamic alpha tuning for hybrid retrieval in retrieval-augmented generation. *arXiv preprint arXiv:2503.23013*.

Z. Zhong, H. Liu, and X. Cui. 2025. Direct retrieval-augmented optimization: Synergizing knowledge selection and answer generation. *arXiv preprint arXiv:2505.03075*.

Appendix

Prompt for LLM-as-a-Judge

Prompt for Extracting Scenarios

You will be given a question and its ground truth answer list where each item can be a ground truth answer...
Here is the criteria for the judgement:

- The `pred_answer` doesn't need to be exactly the same as any of the ground truth answers, but should be semantically the same for the question.
- Each item in the ground truth answer list can be viewed as a ground truth answer for the question, and the `pred_answer` should be semantically the same as at least one of them.

Input format:

```
question: {question}
ground truth answers: {gt_answer}
pred_answer: {pred_answer}
```

Hybrid RAG Case Study (TriviaQA)

Hybrid RAG Case Study

Question ID: `sfq_648`

Question: Which Cypriot born, Greek general led the guerrilla organisation, EOKA, in Cyprus, during the 1950's?

Ground Truth Answer: `George Grivas`

Generated Answer: Based on the provided documents...

Final Score (LLMJ): `5.0`

Final Weights: `Sparse = 0.1954, Dense = 0.8046`