Open Domain Question Answering

Project Guide: Dr. Rajendra Prasath

Navadeep, Chakradhar, Arun

Project code: B25RP03

Open Domain Question Answering

Project Guide: Dr. Rajendra Prasath

Navadeep, Chakradhar, Arun

Project code: B25RP03

Introduction

A system that answers questions without domain restrictions using information from a large corpus

How It Works:

Question Understanding - Preprocessing, Normalization, Semantic understanding

Document Retrieval - BM25, DPR, Hybrid Retrieval

Passage Ranking - TF-IDF Scoring, BERT based models.

Answer Extraction - Extractive Models(BERT) , Generative Models(GPT)

Key Components:

Retriever

Reader

Knowledge Source

Introduction

- Existing Paradigms:
  - -Retrieve-then-read: Relies on external corpora (e.g., Wikipedia).
  - -Generate-then-read: Uses LLMs to generate virtual documents.
- Limitations:
  - -Retrieve-then-read may miss diverse evidence.
  - -Generate-then-read may lack factual accuracy.

Recent works:

1) **Retrieve-then-Read Paradigm**

- Two-stage pipeline: Retriever fetches documents; Reader generates answers.
- Sparse (exact-match): BM25 [1]
- Dense (learned): DPR [2], RocketQA [3], ColBERT [4], ART [5]
- Readers: T5 [6], InstructGPT [7]

2) **Generate-then-Read Paradigm**

- LLMs as generative retrievers [9, 10, 11].
- Generative Retrieval: LLMs create pseudo-documents [8]
- GenRead: Cluster-based evidence generation [9]

Recent works:

3) **LLM Capabilities**

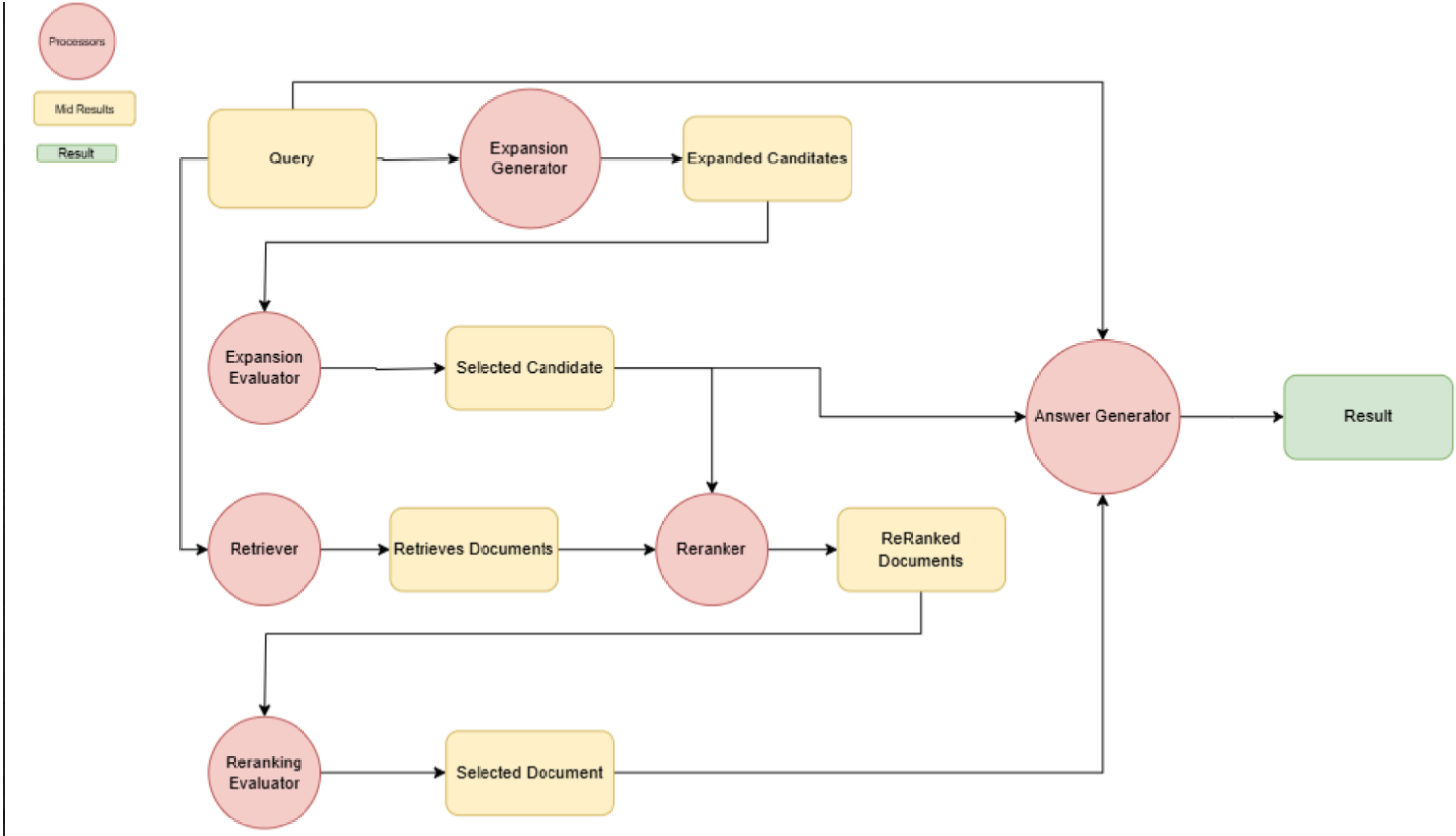
- Few/Zero-Shot Generation: GPT-3 [10], PaLM [11]
- Self-Verification/Refinement: Reflexion [12]
- LLM Reranking: Listwise scoring [13]

4) **Prompt Optimization**

Improving how we ask questions to get better answers from LLMs.

- APE: Iterative prompt search[14] (Tries many prompts and picks the best one)
- DLN: Learnable-prompt parameters [15] (Learns the best prompts automatically.)

Architecture



Proposed Framework

- Three-Step Process:

  -Query Expansion: Generate background context using LLMs.

  -Document Selection: Retrieve and rerank documents.

  -Answer Generation: Produce the final answer.

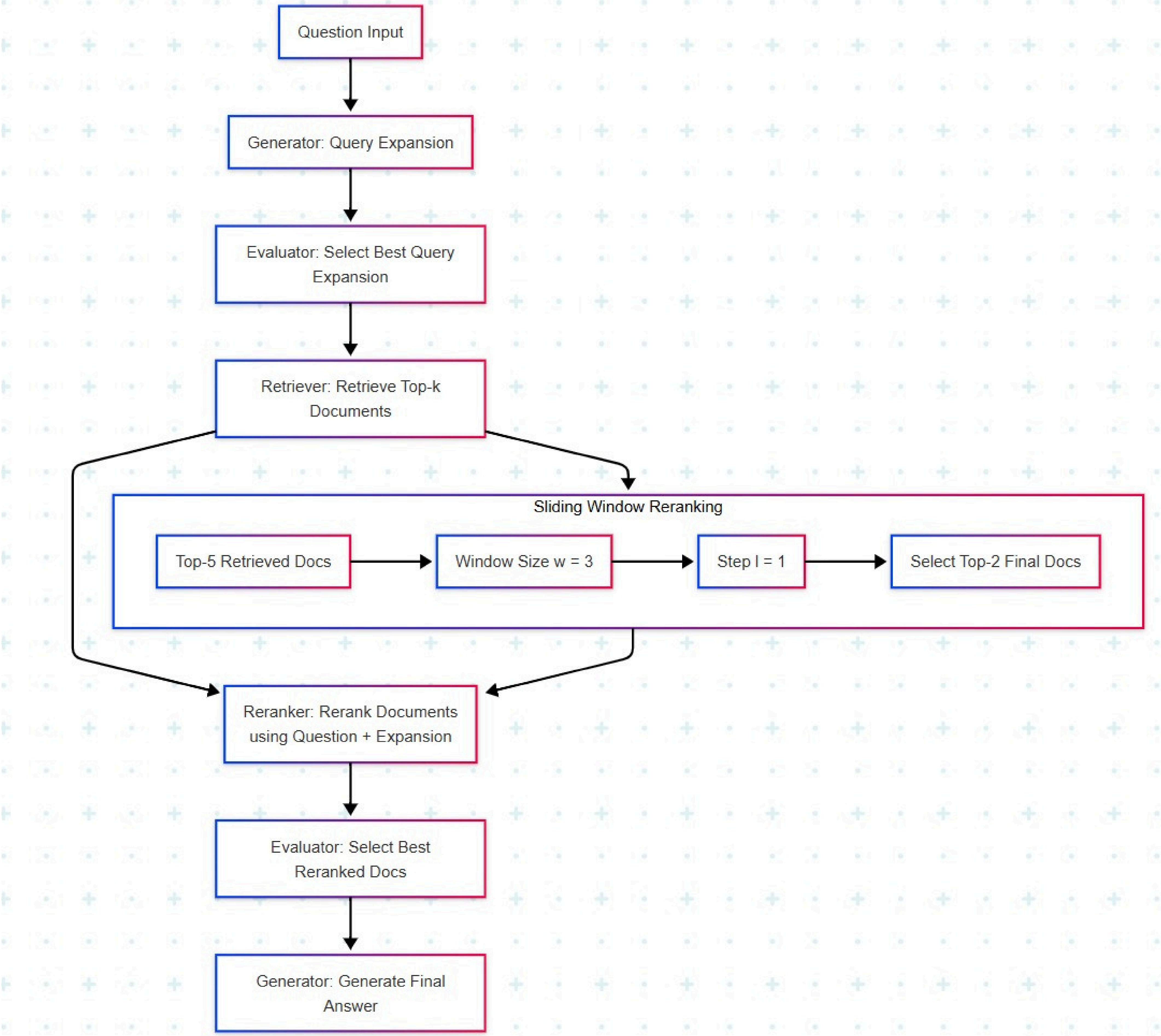
- Multi-Role LLMs:

  -Generator: Expands queries.

  -Reranker: Prioritizes documents.

  -Evaluator: Refines outputs via feedback.

Pipeline



Question Input
Generator: Query Expansion
Evaluator: Select Best Query Expansion
Retriever: Retrieve Top-k Documents
Sliding Window Reranking
Top-5 Retrieved Docs
Window Size w = 3
Step l = 1
Select Top-2 Final Docs
Reranker: Rerank Documents using Question + Expansion
Evaluator: Select Best Reranked Docs
Generator: Generate Final Answer

Methodology Details

Query expansion:

1. Generate background context using LLMs.
    - Example:
        - Question: "Who invented the telephone?"
        - Expansion: "Background: The telephone was patented in 1876. Key inventors include Alexander Graham Bell and Elisha Gray."

2. LLM as Generator (G_e):

$$e = G_e(q;\theta_e)$$

    - Input: Question (q) + prompt (θ_e).
    - Output: Expansion (e) with key details.

Methodology Details

Document selection:

1. Process:
   - question("When was the first iPhone released?")
   - Fine Reranking: LLM reranks top-k docs (k=10) via sliding window (window=20, step=10).

     -"The first iPhone debuted on June 29, 2007." (Score: 0.98)

     -"Apple Inc. was founded in 1976." (Score: 0.45)
2. LLM as Reranker (Rd):

   $$d = Rd(q, e; \theta d)$$

   - Input: Question (q) + Expansion (e) + Optimized prompt (θd).
   - Output: Top-k reranked documents (d).

Methodology Details

Answer Generation:

1. Inputs:
   - Question (q): "Who invented the telephone?"
   - Expansion (e): "Background: The telephone was patented in 1876. Key inventors include Alexander Graham Bell and Elisha Gray."
   - Reranked Docs (d):
     a. "Alexander Graham Bell was awarded the first US patent for the telephone in 1876."
     b. "Elisha Gray filed a patent caveat the same day but lost the legal battle."
2. LLM as Reader ($\boldsymbol{Ga}$):

$$a = Ga(q, e, d; \theta a)$$

   - Prompt (θa): "Summarize the most accurate answer from the evidence."
   - Output (a): "Alexander Graham Bell invented the telephone, patented in 1876."

Output from Our Implementation

Query expansion:

Question: "what did james k polk do before he was president?"

"expansion_candidates":
["<think> Okay, so I need to figure out what James K. Polk did before he became president. I remember that he was a U.S. President, probably in the 19th century, but I'm not exactly sure of the details. Let me start by recalling what I know about him. I think he ....

"expansion_scores": [
0.6969156523537555,
0.6915215483794672,
0.6989462438269207,
0.6916691048144472,
0.6928344002802335,
0.6957589557604544,
0.6937911983566307,
0.687763436870264,
0.6911557235482523,
0.6908004897383319],

Like this 10 passages are generated, then expansion scores are calculated.

LLM Used: deepseek-r1-distill-llama-70b
Embedding model that we used: all-MiniLM-L6-v2

Output from Our Implementation

Document selection:

"ctxs": [
  {"text": "Entity: James K. Polk (ID: m.042f1)"},
  {"text": "Entity: United States Representative (ID: m.02_bcst)"},
  {"text": "Entity: Governor of Tennessee (ID: m.04x_n9q)"},
  {"text": "Entity: Speaker of the United States House of Representatives (ID: m.0cgqx)"}
]

"reranking_scores": [0.569273014141989, 0.569273014141989, 0.569273115222647]

Output from Our Implementation

Answer Generation:

"answers": [
  "Speaker of the United States House of Representatives (ID: m.0cgqx)",
  "Governor of Tennessee (ID: m.04x_n9q)",
  "United States Representative (ID: m.02_bcst)"
]

Experimental Results

- Datasets:
  -WebQsp
- Metrics:
  -Recall, precision, f1

Recall Metrics:

Top-k	Recall(%)
Top-2	63.20
Top-4	83.15
Top-8	91.24

Precision Metrics:

Top-k	Precision(%)
Top-2	49.73
Top-4	46.15
Top-8	33.20

F1 Score Metrics:

Top-k	F1 Score(%)
Top-2	48.89
Top-4	49.24
Top-8	38.26

Comparision with other results

Our work's results surpassed the results existing methods:

On WebQ dataset:

Recall:

Method	**Top-2**	**Top-4**	**Top-8**
GenRead-sampling [9]	58.02	64.67	69.59
GenRead-clustering [9]	61.17	67.47	72.00
Our Implementation	63.20	83.15	91.24

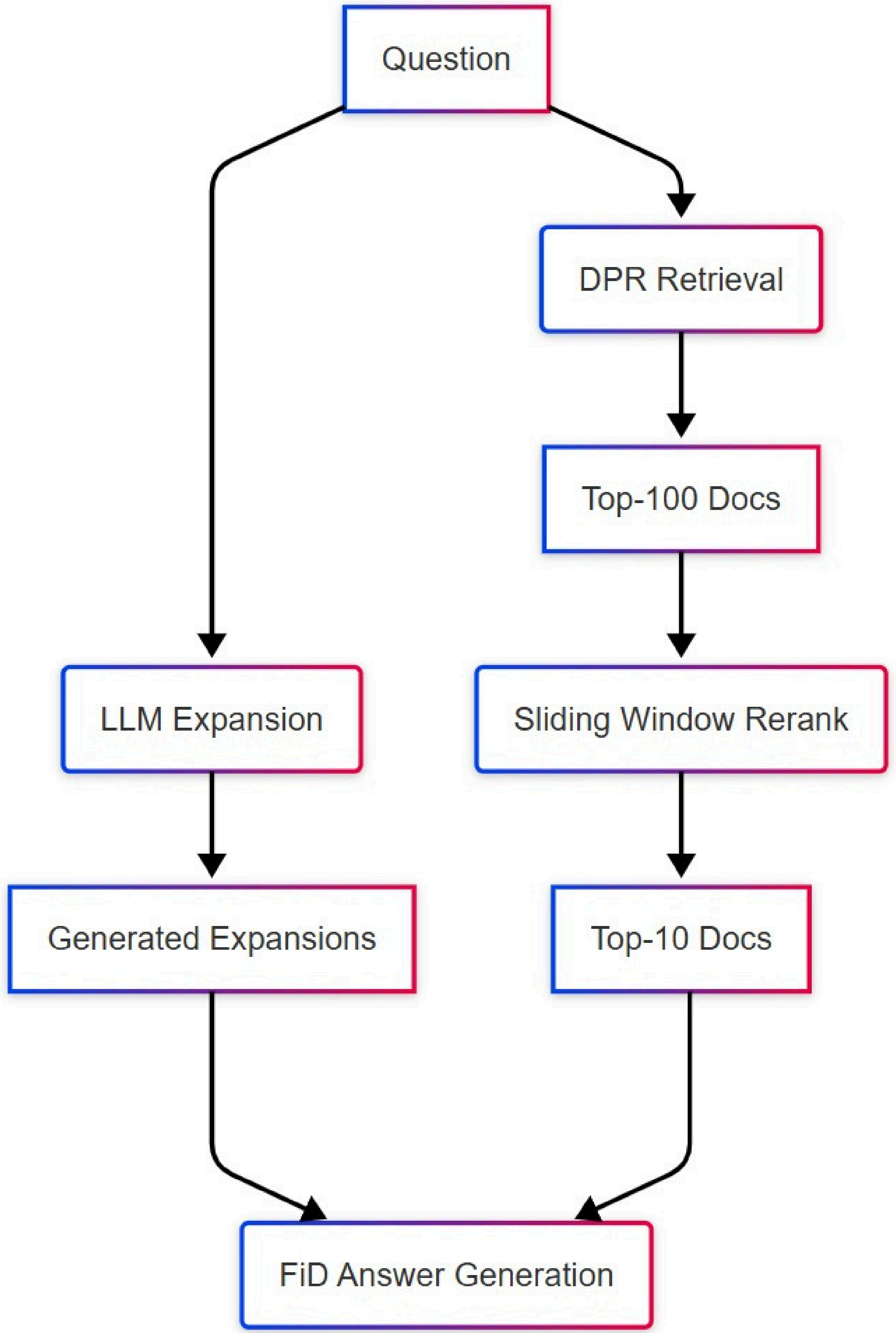
Future Work

Hybrid Retrieval-Augmented Generation (RAG)

Current system uses only LLM-generated expansions.

Combine LLM-generated virtual docs with retrieved evidence to enhance accuracy & coverage.

Retrieve passages from large knowledge sources (e.g., Wikipedia) using sparse (BM25) or dense (DPR) retrievers to ensure factual accuracy.



Future Work

Prompt Optimzation algorithm: [14],[15]

- Goal: Automatically refine prompts for better performance.
- Approach:
  -Treat prompts as learnable parameters.
  -Use variational inference to optimize prompts.
  -Update prompts based on evaluator feedback.

Example:

Step	Initial Prompt	Optimized Prompt
Query Expansion	Generate a document about X.	List key facts, dates, and people related to X.
Document Reranking	Rank these by relevance.	Prioritize docs with direct quotes or numbers.

Future Work

- Test the Impact of Components:

  -Without generator

  -Without reranker

  -Without evaluator

  -Without prompt optimization

- Evaluate it on different ODQA datasets:

  - Currently evaluated on only Webqsp.
  - Evaluate it on NQ, TriviaQA

References:

1. Robertson, S. & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in IR, 3(4), 333–389.
2. Karpukhin, V., Oğuz, B., Min, S., et al. (2020). Dense Passage Retrieval for Open-Domain QA. arXiv:2004.04906.
3. Qu, Y., Ding, Y., Liu, J., et al. (2020). RocketQA: An optimized training approach. arXiv:2010.08191.
4. Khattab, O., Potts, C. & Zaharia, M. (2021). Relevance-guided supervision for OpenQA with ColBERT. TACL, 9, 929–944.
5. Sachan, D. S., Lewis, M., Yogatama, D., et al. (2023). Questions Are All You Need to Train a DPR. TACL, 11, 600–616.
6. Raffel, C., Shazeer, N., Roberts, A., et al. (2020). Exploring the Limits of Transfer Learning with Text-to-Text T5. JMLR, 21(1), 5485–5551.
7. Ouyang, L., Wu, J., Jiang, X., et al. (2022). Training LMs to Follow Instructions with Human Feedback. NeurIPS 35, 27730–27744.
8. Petroni, F., Rocktäschel, T., Lewis, P., et al. (2019). Language Models as Knowledge Bases? arXiv:1909.01066.
9. Yu, W., Iter, D., Wang, S., et al. (2023). GenRead: Generative Retrieval for ODQA. arXiv:2305.xxxxx.
10. Brown, T., Mann, B., Ryder, N., et al. (2020). Language Models Are Few-Shot Learners. NeurIPS 33, 1877–1901.

References:

11) Chowdhery, A., Narang, S., Devlin, J., et al. (2022). PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311.
12) Shinn, N., Cassano, F., Labash, B., et al. (2023). Reflexion: Language Agents with Verbal RL. arXiv:2303.11366.
13) Ma, X., Zhang, X., Pradeep, R. & Lin, J. (2023). Zero-Shot Listwise Document Reranking. arXiv:2305.02156.
14) Zhou, J., Chen, X., Liu, Y., et al. (2023). APE: Automatic Prompt Engineering. Under review.
15) Sordoni, A., Yuan, X., Côté, M.-A., et al. (2023). Deep Language Networks: Joint Prompt Training. arXiv:2306.12509.

Thank you