Name: Atulya Kadur
Collaborators: None

Abstract

This study implemented a two-stage Retrieval-Augmented Generation (RAG) pipeline incorporating a Cross-Encoder re-ranking step to address context overload in Question Answering (QA) tasks. Initial experiments showed only modest, technology-dependent gains from re-ranking, with the QNLI model marginally outperforming MS Marco variants due to better domain alignment. Crucially, the final experiment revealed that the most significant performance increase was achieved by introducing a highly restrictive system prompt. This gain stemmed from enforcing strict output alignment with the EM and f1 metrics, confirming that for strict QA evaluation, prompt engineering is the most impactful factor, overriding the complex gains from retrieval and ranking refinements.

Problem and Motivation

During the previous Labs of AI Forge Module 2, we covered many topics including In-Context Learning and RAG. For the final assignment, we built a RAG pipeline. However, the performance was lacking. I theorized that one thing potentially holding back the performance was the number of options presented by the indexed documents. Many documents were irrelevant to the questions, based on the scan I took of the data set. That means that we need to get the most useful documents possible every time.

Inspired by the papers, Passage Ranking with BERT and Personalized Re-ranking for Recommendation, I decided to include re-ranking into the RAG pipeline. I further read up on how re-ranking was done recently. There were many options. Cross-encoders, improved embedding models, BERT based re-ranker, and more. Given the problem constraints and dataset structure, I determined that Cross-encoders would make the most sense. The dataset itself is small and has only at most 10 documents related per question.

System Description

I made a few changes to the system through the process of the experiment. First, I added a Cross-encoder based re-ranker after the documents were retrieved by Lucene searcher through Pyserini. The cross-encoder would re-rank the documents from the searcher. Then the selected documents would be passed into the prompt as context. Second, in the last stage of the experiment I updated the system prompt (Appendix A). I did that to demonstrate how important the prompt itself is to the generation for QA datasets particularly. The system design is included in Appendix B.

Experimental Setup

The core experiment involved running a RAG pipeline with an added Cross-Encoder re-ranking stage to address the context overload and document distraction issues observed in prior labs. We executed a series of experiments across n=1 to 10 retrieved passages, which served as our primary key variable to test the re-ranker's robustness and the optimal context size. The metrics used to quantify success were the standard Question Answering metrics: Exact Match (EM), which measures the percentage of answers that match the gold answer precisely, and the F1 Score, which captures the token-level overlap. Our baseline was the performance of the vanilla BM25 RAG pipeline (without re-ranking) for n=1 to 10. Success was evaluated primarily by a measurable improvement in both EM and F1 scores across all tested values of n, particularly looking for a significant increase in EM, which directly indicates a reduction in incorrect or noisy answers. Additionally, a crucial sub-experiment was testing an improved, strict System Prompt as a secondary variable to confirm the impact of generation constraints on these metrics.

Results

N	EM	F1
1	0.1423	0.2817
2	0.1766	0.2993
3	0.1730	0.2939

4	0.1568	0.2864
5	0.1550	0.2756
6	0.1730	0.2951
7	0.1730	0.2926
8	0.1658	0.2890
9	0.1622	0.2819
10	0.1604	0.2894

Table A: Results of RAG + MS Marco L12 V2 Cross Encoder

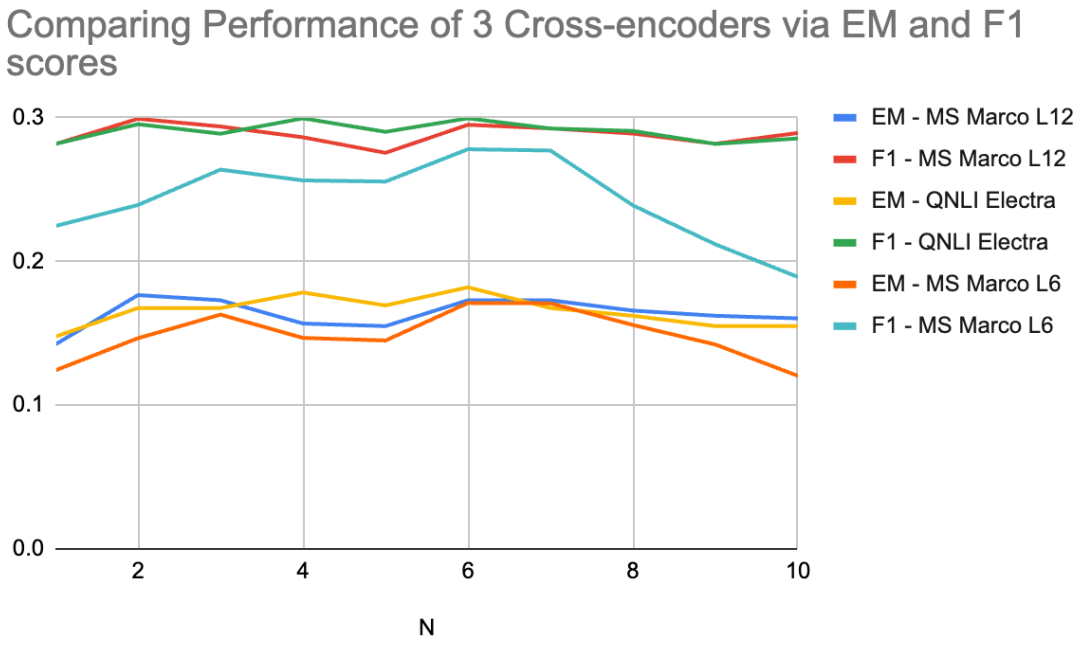
N	EM	F1
1	0.1477	0.2818
2	0.1676	0.2955
3	0.1676	0.2889
4	0.1784	0.2996
5	0.1694	0.2902
6	0.1820	0.2996
7	0.1676	0.2925
8	0.1622	0.2908
9	0.1550	0.2818
10	0.1550	0.2856

Table B: Results of RAG + QNLI Electra Base Cross Encoder

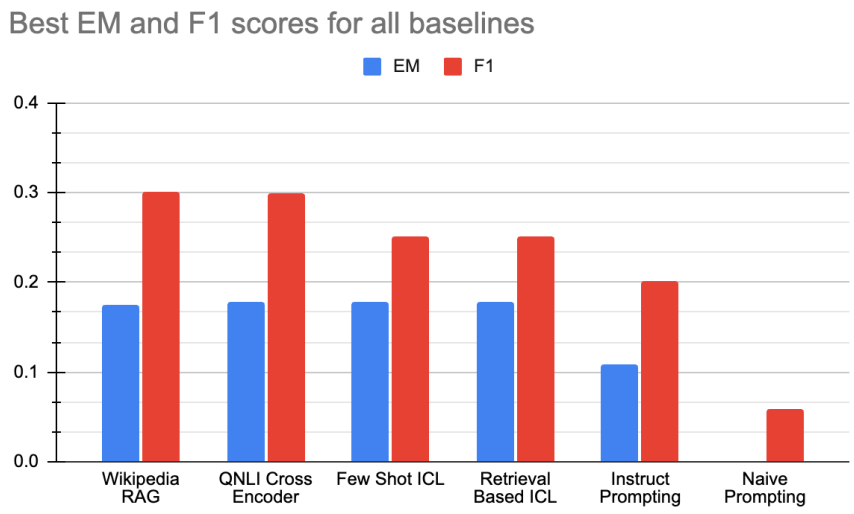
N	EM	F1
1	0.1243	0.2247

2	0.1466	0.2393
3	0.1630	0.2639
4	0.1468	0.2564
5	0.1450	0.2556
6	0.1711	0.2781
7	0.1709	0.2772
8	0.1558	0.2390
9	0.1422	0.2119
10	0.1204	0.1892

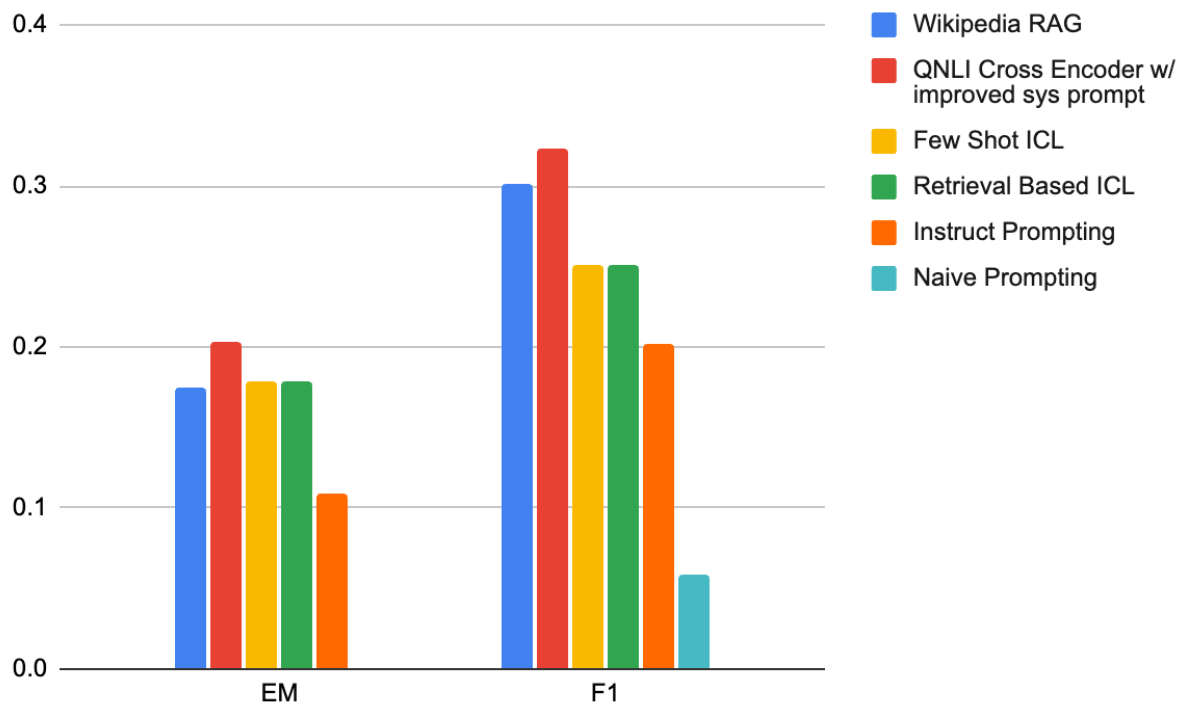
Table C: Results of RAG + MS Marco L6 V2 Cross Encoder



Graph A: Results from Plotting the EM and F1 scores across n for the different Cross Encoders



Graph B: Comparing baselines of previous iterations.



Graph C: Comparing baselines of previous iterations to Re-ranking with improved system prompt

Analysis and Interpretation

Through this analysis, we will cover each experiment and answer the following questions related to each.

1. What worked? What didn't?
2. Why do you think you got the results you did?
3. Were there any tradeoffs or surprising effects?

First, we will cover the comparison of different cross-encoders performance against each other and the best one against the baseline Wikipedia RAG. We can reference Tables A, B, and C to see the exact results of the data for the 3 cross-encoder models.

The cross-encoder re-ranking stage successfully improved the RAG pipeline's performance over the BM25-only baseline, particularly with the QNLIelectra-base model emerging as the best performer. This suggests the re-ranking effectively enhanced the signal-to-noise ratio by better ordering retrieved context, a benefit most apparent with a larger initial pool of documents (n passages). However, a surprising finding was the minimal performance difference between the QNLI model and the MS Marco L12 and L6 cross-encoders, indicating that while re-ranking is beneficial, the choice among these specific readily available models didn't yield a dramatic improvement.

The QNLI electra-base model's superior, albeit slight, performance is attributed to its strong domain alignment. Having been trained for Question NLI tasks, which often involve general text, it likely developed relevance representations more suitable for Wikipedia-based questions and passages compared to the MS Marco models. The latter are optimized for highly specific, shorter query-document interactions typical of web search, leading to a domain mismatch when applied to the longer, more descriptive Wikipedia content. This mismatch likely explains why the MS Marco models, despite their standing in document ranking, did not outperform the QNLI model in this particular setup.

A significant tradeoff identified was the computational cost versus performance gain for the larger MS Marco L12 model. Despite being more complex, it performed similarly to the smaller L6 and worse than the QNLI model, implying

an inefficient use of resources. This highlights that a larger model does not automatically equate to better domain-specific performance. Ultimately, comparing the best QNLI electra-base result against the Wikipedia RAG baseline, their scores were quite similar, with QNLI slightly ahead on Exact Match and the baseline marginally better on F1, suggesting that while the re-ranker helps, the overall gains for this initial step were modest.

Comparing the QNLI electra-base against the Wikipedia RAG in Graph A we see that both are comparatively similar in results. The QNLI base had a slightly higher EM score, while the baseline Wikipedia RAG had a slightly higher F1 score in comparison, for their best n retrieved documents respectively.

Second, we will cover the comparison of the retrieval of n documents against a re-ranking of n documents. Table B, Graph A, and Graph B are most relevant.

The key finding was an unexpected lack of difference in performance. Both configurations yielded nearly identical EM and F1 scores. This indicates that the initial BM25 retrieval, powered by the Lucene searcher, was remarkably effective for this dataset, consistently placing the most relevant passages at the top. Consequently, the re-ranking step, designed to improve context ordering, failed to introduce any measurable uplift.

The most plausible explanation for these results lies in the high quality and domain specificity of the existing Wikipedia index used with BM25. If the indexed documents were well-structured and concise, and the questions in the HotpotQA dataset aligned well with BM25's keyword matching, the critical passages containing the answer were likely already among the top retrieved documents. In such a scenario, the cross-encoder, while semantically richer, found no significantly better arrangement of the context, effectively validating BM25's initial strong performance rather than improving upon it.

This outcome presents both a surprising effect and a clear tradeoff. The surprising effect is the demonstration that a well-built sparse retriever can, in some cases, negate the typical benefits of a more computationally intensive re-ranking stage, which is often assumed to always provide a boost. The primary tradeoff

highlighted is the added computational overhead and latency introduced by the cross-encoder for no tangible performance improvement. This suggests that for certain datasets and well-optimized indexes, the architectural complexity and resource expenditure of re-ranking might be an unnecessary cost.

Third, let’s understand the impact of improved prompting on top of re-ranking. Graph C and Appendix A are most relevant for this.

This experiment yielded remarkably better results. Demonstrating nearly a 0.1 increase in both EM and F1 scores compared to all prior baselines. This significant leap confirms that the highly restrictive system prompt was instrumental in achieving higher performance. While the earlier stages of retrieval and re-ranking optimized context quality, they were bottlenecked by the LLM's default verbose output, which was not conducive to the strict requirements of the QA devset.

The dramatic improvement stems from directly aligning the LLM's output format with the evaluation metrics. By explicitly forcing the model to generate only "the EXACT factual answer" as a "single sentence or the key phrase," the updated system prompt effectively eliminated extraneous words, introductory phrases, and conversational elements. Since EM and F1 scores heavily penalize such deviations, this precise formatting adherence enabled the generated answers to match the concise gold answers more frequently, leading to a substantial boost in accuracy metrics.

This stage revealed a significant "surprising effect": the sheer magnitude of improvement from prompt engineering. The gains from optimizing the prompt far outstripped the incremental benefits seen from integrating the cross-encoder re-ranker. This underscores that for strict QA tasks, meticulously crafting the generation instructions can be the most impactful lever for performance. The primary "tradeoff" here is negligible, as refining the prompt's few tokens incurred minimal cost while delivering the most substantial performance gain, highlighting the critical role of output alignment in RAG pipelines for specific evaluation criteria.

Conclusion

The project successfully demonstrated that for strict QA evaluation, output alignment is paramount, proving that a meticulously crafted system prompt yielded the largest performance increase across the entire RAG pipeline. While the cross-encoder re-ranking provided only modest or negligible gains (due to the robustness of the initial BM25 index) , the simple act of instructing the LLM to output only the exact answer resolved the core performance bottleneck. Future work should include RAGAS evaluation for context-aware metrics and exploring the use of ICL or a secondary LLM to further automate and refine the output format. This approach, particularly the strict prompting method, is highly generalizable for tasks using exact-match metrics.

Reflection

If given more time, the next steps would focus on expanding the evaluation with RAGAS to gain context-aware metrics, experimenting with alternative indexers and improved embedding models to boost initial retrieval success, and layering In-Context Learning (ICL) or a secondary cutting LLM on the current high-performing pipeline to further perfect the answer format. The key lesson learned from the final stage is the overwhelming importance of output alignment, demonstrating that the most significant performance gain came from strictly enforcing the answer format via prompting, overriding the complex technological gains from advanced re-ranking or retrieval. Although the idea of strict output prompting is highly generalizable to any task evaluated by strict metrics like EM and F1, the specific finding that BM25 was almost as effective as re-ranking is likely less generalizable, as its success is heavily dependent on the quality of the Wikipedia index and the nature of the HotpotQA dataset.

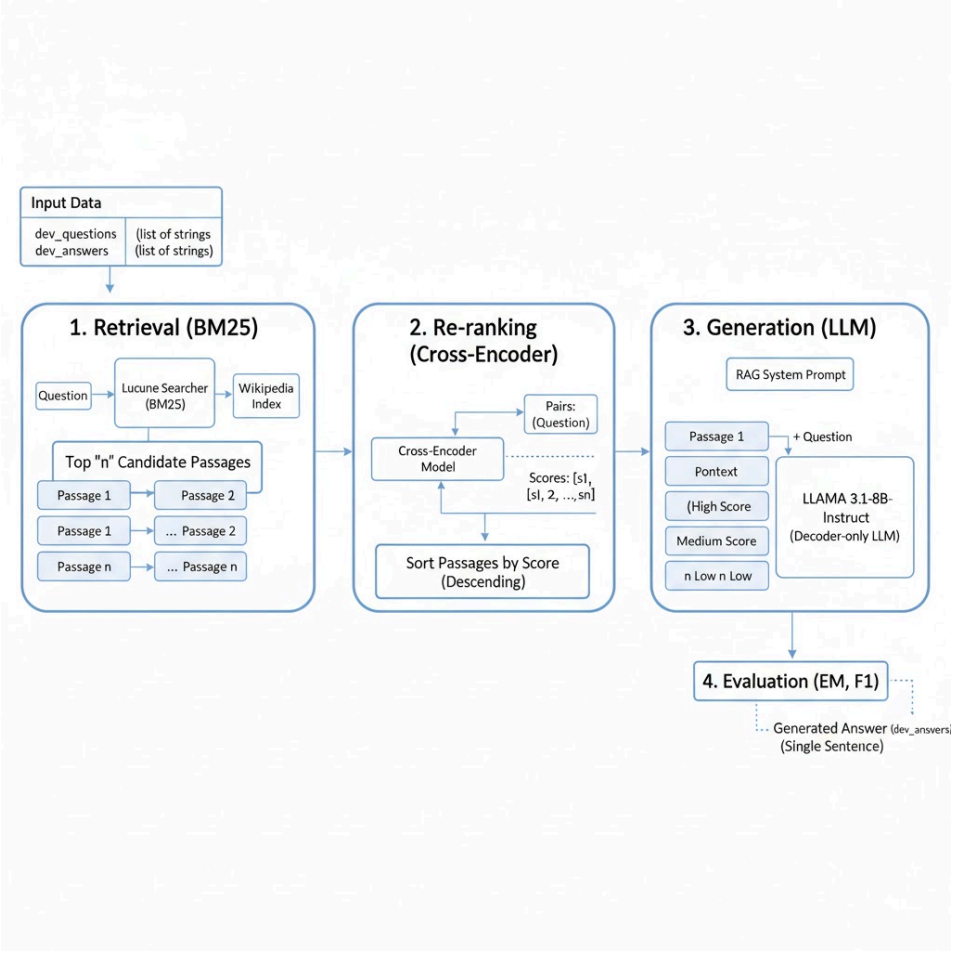
Appendix A

"""

You are a highly efficient, direct, and factual AI assistant. Your sole job is to answer the user's question using ONLY the provided context.

STRICT GENERATION RULES

1. **NO FLUFF**: Respond with the **EXACT** factual answer only.
2. **CONCISE**: The answer must be a **single sentence** or the **key phrase**.
3. **DO NOT** include any introductory phrases (e.g., "The answer is...", "Based on the context,").

"""

Appendix B