Enhancing Q&A Text Retrieval with Ranking Models: Benchmarking, fine-tuning and deploying Rerankers for RAG

Summarization of key points

Basic introduction of the paper

Ranking models are like helpers that make sure we find the best answers when we search for something. When you ask a question, these models look through lots of information and pick out the most important pieces, putting them in the right order so you can see the best answers first. They help make searching for information easier and more accurate!

This paper looks at different ways to help computers find the right answers to questions. It checks how well these ways work and if they can be used in real-life situations, like in businesses. The goal is to make sure the computer can find the best answers quickly and accurately.

This paper has created a new model called **NV-RerankQA-Mistral-4B-v3** which is basically a ranking model which has a 14% increase in accuracy compared to previous **v1** and other tools that are available

1)Introduction to text retrieval systems

There are basically two models that need to be focussed in text retrieval systems

1) Embedding models
2) Ranking models

Embedding model

- These are models which find and understand text better, it does this by converting the text document by converting them into vectors where It uses the transformer architecture to do so
- Some examples of the Embedding models are Sentence-BERT and E5
- These models learn by comparing questions to answers to make sure they find the most relevant information and give the answer to the given question.
- Retrieval systems that use embedding models break down large amounts of text into smaller pieces like sentences and paragraph and then these pieces get turned into a special format called embeddings and store them in a database
- This makes it easier to search the right pieces of text that answer a question by using MIPS(Maximum inner product search) or approximate nearest neighbor(ANN)

Ranking models : These are in simple words a special computer program where its main job is to

- To find the best answers by looking at both questions and answers together
- It uses a method called self attention to understand how the question and answer relate to each other
- By using these models we can accurate results without very large and expensive programs

Text retrieval has been incorporated into **Retrieval-Augmented Generation (RAG)** systems, enabling large language models (LLMs) to leverage external knowledge.

2) Multi-Stage retrieval pipelines it involves two key stages

1) Candidate retrieval :Embedding models are used to retrieve top-k passages relevant to a query.
2) Reranking: A ranking model is used to reorder the retrieved passages based on their relevance to the query.

Ranking models (e.g., cross-encoders) are typically used in the second stage to improve retrieval accuracy by deeply modeling the relationship between the query and passages using self-attention mechanisms.

3) Benchmarking models :

1) This paper assess the various embedding and ranking models based on the Q&A dataset

2) NV-RerankQA-Mistral-4B-v3 is introduced in this paper, this is a state of the arc ranking model which has a 14% increase in accuracy compared to other rerankers

3) This paper shows a possibility of using small embedding models and the proposed reranker model can reduce the indexing time and computational while giving a high accuracy

4) key models that are being used and their purpose :

1) Embedding models
    a) Snowflake-arctic-embedded : it is a bert based model which has around 335 million parameters so this model is reasonably expensive and can be used

    b) nvidia**/**nv-embed-e5-v5: like the previous model this model also has 335 million parameters however this model is mainly designed for Q&A tasks (Question and answering)

    c) nvidia/nv-embed a-mistral-7b-v2 :This is the largest embedding model with over 7 billion parameters making it the most computationally expensive model to train this model is capable of capturing more complicated relationships between texts

2) Reranking models
    a) ms-marco-MiniLM-L-12-v2 :Lightweight ranking model with 33 million parameters.Fast and suitable for low-latency applications.
    b) jina-reranker-v2-base-multilingual : 278 million parameters, providing a balance between efficiency and accuracy.
    c) mixed breed-ai/mai-rerank-large-v1: Provides high accuracy in tasks requiring nuanced understanding between queries and passages.More computationally expensive than smaller models.
    d) Bge-reranker-v2-m3: 568 million parameters, suitable for tasks requiring high accuracy and handling multiple languages.enhancing performance for multilingual tasks.
    e) NV-RerankQA-Mistral-4B-v3:The paper introduces NV-RerankQA-Mistral-4B-v3, a large and powerful reranking model with 4 billion parameters. It is fine-tuned from the Mistral 7B architecture by pruning it down to 4 billion parameters to reduce its computational footprint while maintaining high accuracy.
    This model is designed specifically for question-answering tasks, using contrastive learning to optimize its performance in distinguishing between relevant and irrelevant passages.
    The bi-directional attention mechanism is employed, allowing the model to look at the entire context when reranking passages, which improves accuracy.

    According to the benchmarks, it outperforms all other reranking models, achieving a 14% accuracy improvement over the next-best reranker, making it highly suitable for commercial and industrial applications where high precision is critical.

5) Evaluation and comparison :
The paper evaluates these embedding and ranking models on Q&A datasets from the BEIR benchmark (e.g., Natural Questions, HotpotQA, and FiQA)
using the NDCG@10 metric

The results show that

1) Smaller embedding models like the snowflake model and the nv-embeda-e5-qa benefit from the proposed reranking by cross encoding NV-RerankQA-Mistral-4B-v3
2) The NV-RerankQA-Mistral-4B-v3 consistently outperforms other models across all datasets and embedding model combinations, achieving the highest retrieval accuracy.

6) Conclusion :
1) Embedding models like snowflake-arctic-embed-l and nv-embedqa-e5-v5 are smaller, more efficient options, while models like nv-embed a-mistral-7b-v2 provide greater accuracy at a higher computational cost.

2) Ranking models are critical for improving retrieval accuracy. Lightweight models like ms-marco-MiniLM-L-12-v2 are efficient, but models like **NV-RerankQA-Mistral-4B-v3** dominate in terms of performance, making them the best choice for high-stakes applications requiring precision in information retrieval.