Aidan Bell

12/05/24

Github: https://github.com/aidan073/IR-Assignment5

LLM Query Expansion Search Experiments

Initial (Discarded) Approach

My initial idea was to prompt an LLM for query expansion, and use the expanded queries with a bi-encoder. I explored several flan-t5 models (ranging from t5-small to t5-xl). Despite being efficient, pre-trained flan-t5 models generated disappointing results. T5-small through t5-large produced unpredictable, poor quality expanded queries. I will refer to a prompt as Q2D/CoT, which is outlined in this (Jagerman et al.) paper as: “Answer the following question: {query} Give the rationale before answering”. The resulting LLM generation is then appended to the end of the original query. The research of Jagerman et al. found that flan-t5 Q2D/CoT with BM25 resulted in better performance than original query BM25. Despite their findings, figure 1 shows my best produced results after several different prompts and models, which was still worse than my baseline in figure 2 (non-expanded queries). The final prompt I used to achieve the results in figure 1 is outlined in this (Rizzo et al.) paper as Q2D/ZS, which is as follows: “Write a passage that answers the following query: {q}”. Again, the resulting LLM generation is appended to the end of the original query. Q2D/CoT performed slightly worse than Q2D/ZS for me, supporting the results of Rizzo et al., and contrasting with Jagerman et al. I subsequently attempted to perform document expansion with a fine-tuned flan-t5 model as provided by this repo, however it was far too slow on available hardware.

Metric	P@1	P@5	nDCG@5	MRR	MAP
flan-t5-xl	0.689	0.476	0.498	0.786	0.450

Figure 1. flan-t5-xl Expanded Queries with Bi-Encoder Search

Metric	P@1	P@5	nDCG@5	MRR	MAP
Baseline	0.703	0.480	0.503	0.796	0.455

Figure 2. Original Queries with Bi-Encoder Search

Further Challenges

LLAMA created what appeared to be context rich and high quality expanded queries, but it had a high latency. Loading the model in 8-bit cut the latency by ~50%, making it more viable, though with a slight loss in precision. I decided on LLAMA3.1-Instruct-8B for query expansion, with hyper-parameters: top_k=20, top_p=0.90, temperature=0.6. I initially had more loose bounds on top_k and top_p, but noticed a strange occurrence of LLAMA adding “Answered by: <name>” to the expanded queries. These instances disappeared upon constraining top_k and top_p. I experimented with two main prompts, the aforementioned Q2D/CoT and Q2D/ZS, and a variety of other custom prompts. I compared two bi-encoders, gte-large-en-v1.5, and multi-qa-MiniLM-L6-cos-v1. Comparing the results in figure 3, gte-large-en-v1.5 provides a large jump in performance. As supported by the research of Rizzo et al., in my experiments, LLAMA expanded queries were better with Q2D/ZS than Q2D/CoT. However, in contrast to the research, LLAMA expanded queries yielded worse results than the original queries (figure 4). I suspect several different reasons behind this discrepancy. First and foremost, the baseline model in Rizzo et al. and Jagerman et al.’s research was BM25. It would make sense that a BM25 model would improve from expanded queries, since it is based upon word matching. However, my baseline is a bi-encoder model using a sentence-transformer. Sentence-transformers won’t necessarily benefit from added words, unless the added words provide additional relevant semantic information. Secondly, even with prompts explicitly telling LLAMA not to generate step by step answers, many expanded queries would contain step by step answers to the original query, which did not mimic the format of the true answers. Finally, LLAMA 3.1 comes with safety features that are likely very harmful to query expansion. Many

expanded queries were appended with a note, detailing the model's neutrality or in-expertise. For example: "The response is written in a neutral and informative tone, without taking a stance or expressing an opinion." These safety features generate copious amounts of noise, which obscures query/doc similarity measurements.

Metric	P@1	P@5	nDCG@5	MRR	MAP
MiniLM-L6	0.740	0.484	0.526	0.825	0.476
gte-large	0.830	0.610	0.635	0.891	0.618

Figure 3. LLAMA3.1 Expanded Queries with Two Different Sentence-Transformers

Metric	P@1	P@5	nDCG@5	MRR	MAP
EQ gte-large	0.830	0.610	0.635	0.891	0.618
OQ gte-large	0.870	0.616	0.646	0.916	0.621

Figure 4. Expanded Queries with Q2D/ZS Prompt (EQ) vs Original Queries (OQ)

Final Model

Underwhelmingly, my final model simply embeds query/doc pairs using NV-Embed-v2, and ranks based on cosine similarity. Massive Text Embedding Benchmark (MTEB) has placed NV-Embed-v2 at the top for retrieval on a public leaderboard. I do believe it is possible to achieve better-than-baseline results with query expansion and bi-encoder ranking. Future work should consider further prompt-tuning to minimize noise/maximize semantic information, and experimenting with different state-of-the-art LLMs.