Enhancing Retrieval via Cognitively Motivated Document Expansion

Giacomo Loss[1] and Andreas Stephan[2] and Matthias Aßenmacher[1,3]

[1]Department of Statistics, LMU Munich,
[2]LDA - Legal Data Analytics GmbH, Munich, Germany,
[3]Munich Center for Machine Learning (MCML), LMU Munich
Correspondence: matthias@stat.uni-muenchen.de

Abstract

This study examines the potential of leveraging large language model (LLM) capabilities to enhance performance in document retrieval tasks. Using human-written prompts based on the 5E Instructional Model from educational psychology, we generate alternative versions of documents in a given corpus using an LLM, tapping into its vast knowledge base. These generated texts can then be used in retrieval tasks, complementing or replacing the original corpus before applying fusion algorithms to combine the results. While the generated texts individually do not outperform the original corpus, fusing retrieval results from multiple generated corpora with those of the original corpus often leads to performance improvements. This suggests that LLM-generated documents, while not a substitute for the original, can complement it to enhance retrieval performance.

1 Introduction

Information is abundant and distributed across various heterogeneous sources, including web pages, digital libraries, and forums. Yet, even when an answer exists, it is often difficult to locate the most relevant information. This is the core problem of Information Retrieval (IR): given a user query and a large document collection, retrieve the most relevant documents. Early IR methods relied on exact term matching, which suffers from the *Vocabulary Mismatch Problem* (Furnas et al., 1987), as they ignore semantic similarity between queries and documents that use different wording.

Recent advances in deep learning have transformed IR and NLP. Word embeddings (Mikolov et al., 2013) and Transformer-based contextual representations (Vaswani et al., 2017) help mitigate vocabulary mismatch by capturing semantic and contextual information. Building on these advances, large language models (LLMs) trained on massive, heterogeneous corpora have proven effective in many tasks, including generation, translation, and dialogue. In this work, we investigate how the knowledge retained by modern LLMs can be leveraged for IR. Specifically, we study their use for *document expansion*—a technique that, alongside the widely adopted *query expansion*, enriches the term space to better match user information needs and improve retrieval effectiveness.

Contributions. Our study has two main goals. First, inspired by the 5E Instructional Framework, we design a set of targeted, cognitively motivated prompts and utilize them with an LLM to generate alternative texts for each document in a corpus, which can also be used for retrieval (cf. Fig. 1). Here, we focus on expanding document corpora in a query-independent manner, leveraging the extensive retained knowledge and generalisation capabilities of LLMs. We thereby contribute to broadening the body of research, as this line of research has received comparatively less attention than, for example, query expansion. Second, we investigate whether combining retrieval results from different generated corpora can enhance retrieval performance. We find that, although individual expansions rarely outperform the original corpus alone, combining retrieval outcomes from multiple generated corpora with those from the original corpus consistently improves effectiveness. This suggests that LLMs can generate complementary views of a corpus, thereby enhancing retrieval performance. Our code is publicly available on GitHub.[1]

2 Background & Related Work

We integrate concepts from educational science (the 5E Instructional Model) with IR techniques (document expansion), leveraging LLMs' broad

[1]https://github.com/GiacoL/Enhancing-Retrieval-via-Cognitively-Motivated-Document-Expansion

Proceedings of the 11th Edition of the Swiss Text Analytics Conference (SwissText 2026) – Scientific Track, pages 18–29

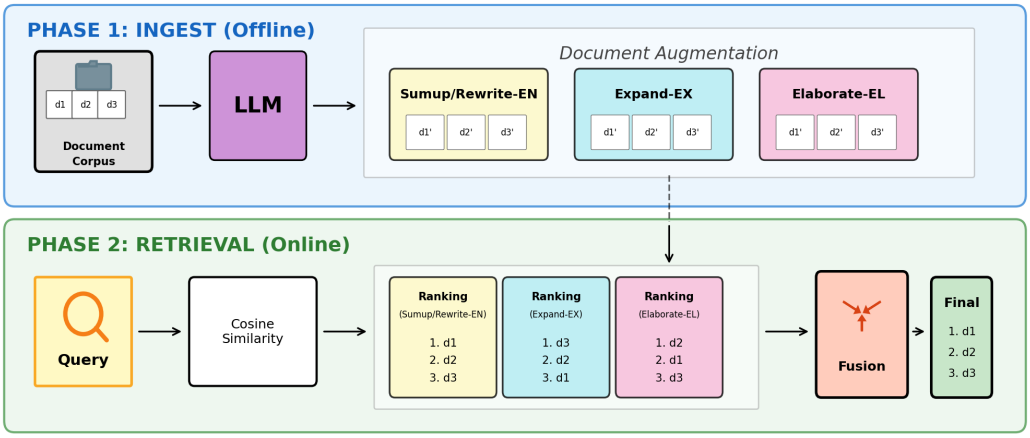

Figure 1: During the **ingestion** phase, each document in the corpus is passed to an LLM using prompts designed to summarize, expand, or elaborate the text. Each prompt yields an alternative version of the corpus, resulting in multiple generated corpora. In the **retrieval** phase, a set of queries is run against each generated corpus, producing a ranking for every (query, corpus variant) pair. Finally, these rankings are fused to obtain a single final document ranking for each query.

knowledge to connect these perspectives. This synthesis provides a coherent structure for organizing and positioning the related work.

2.1 5E Instructional Model

The 5E instructional model (Bybee, 1990), developed by the Biological Sciences Curriculum Study (BSCS), is a pedagogical framework designed to support effective instruction in science education (Duran and Duran, 2004; Joswick and Hulings, 2024). It belongs to the family of constructivist approaches, which posit that learning integrates prior knowledge with hands-on experience. The 5E model comprises five phases: *(i) Engage*: students are presented with new evidence from which a problem emerges and are encouraged to reconcile it with *prior* knowledge; *(ii) Explore*: students *actively* use retained knowledge to make observations and propose solutions to the problems posed by the new evidence; *(iii) Explain*: the teacher formalizes the new concepts in a way that can be assimilated; *(iv) Elaborate*: students apply the new ideas in different contexts and scenarios; *(v) Evaluate*: the focus is on how well students *transfer* concepts to new contexts. As noted by Ruiz-Martín and Bybee (2022), the 5E model rests on recognizable cognitive principles such as the activation of prior knwoledge in the *Engage* phase and its use in the following phase, the *Explore* phase, to find solutions to the problems posed by the new evidence; this aligns with our goal of expanding or shortening corpus documents with an LLM, using a cognitively motivated prompt scheme based on the 5E model, with the LLM acting as the *student*.

2.2 Query & Document Expansion Strategies

Given a large text corpus and a set of queries, the goal of IR is to retrieve the most relevant documents for each query. The process is typically decomposed into two stages. First, in the *retrieval* stage, the relevance of each document to the query is estimated to select the most promising candidates. Second, in the *reranking* stage, these candidates are reordered to refine their relevance-based ranking. This study focuses on the *retrieval* stage: given a set of queries and a text corpus, we aim to retrieve, for each query, the relevant documents.

Early retrieval algorithms were sparse, term-based methods such as *BM25* (Robertson et al., 1995) suffered from the *Vocabulary Mismatch Problem*. Especially after the advent of Transformers (Vaswani et al., 2017), dense retrievers entered the IR landscape (Xiong et al., 2021; Karpukhin et al., 2020). Dense representations of words and sentences enabled a shift from a purely term-based to a more semantic-based setting. However, these dense retrieval systems are often supervised and thus require large amounts of training data, which can be costly to obtain in many scenarios.

A common challenge in IR is that queries are typically short, whereas documents are relatively lengthy and may be written in ways that deviate

substantially from the user query. To mitigate this problem, various *expansion* techniques have been proposed. On the one hand, *query expansion* methods augment queries with external resources or synonymous terms, as in Voorhees (1994). On the other hand, document expansion has also been explored. Tao et al. (2006) or Efron et al. (2012) expand short documents by submitting them as pseudo-queries to a search engine and appending retrieved content; Nogueira et al. (2019b) expand instead documents with queries generated from the documents themselves. Another line of research has focused not on expanding the documents themselves, but on enriching their embedding representations. For example, Zhang et al. (2022) generate multi-view embeddings that can align with different queries, while ColBERT (Khattab and Zaharia, 2020) adopts an interaction-based paradigm to model fine-grained relationships between query and document terms.

2.3 LLMs in IR

A recent line of work has employed LLMs for IR tasks. As documented in the extended survey by Zhu et al. (2023), there are two major approaches to using LLMs for retrieval. On the one hand, LLMs and their extensive retained knowledge can be exploited to reformulate documents or queries to better align with the user's intent. On the other hand, they can be used for data augmentation, for example, by acting as a substitute for human annotators to provide pseudo-relevance feedback. According to Zhu et al. (2023), these models can thus enhance existing retrieval methods and enable new developments via *in-context learning*, which improves retrieval performance by leveraging only a few examples.

So far, most attention has been devoted to query expansion or rewriting with LLMs: Wang et al. (2023) generate pseudo-documents with an LLM and concatenate them to the original query, while Jagerman et al. (2023) expand queries by exploring different zero-shot, few-shot, and chain-of-thought prompting strategies. More recently, Xia et al. (2025) combine LLMs with knowledge graphs to jointly model semantic and structural relations between documents and queries.

By contrast, *document expansion* has received comparatively less attention, likely due to the substantial computational cost of expanding large document collections. Nogueira et al. (2019b) and Nogueira et al. (2019a) use LLMs to predict queries relevant to a document, which are then appended to the document to create an expanded corpus. Gao et al. (2023) generate hypothetical documents with an LLM given a query and use their vector encodings to retrieve relevant documents from the corpus. Yu et al. (2022) generate contextual documents for a given query, whereas Bonifacio et al. (2022) generate queries from documents to construct query–document training pairs for dense retrievers. Jeong et al. (2021) expand the text corpus with additional sentences generated by a pretrained LLM, in a manner similar to our approach.

Although expansion appears beneficial for retrieval, Weller et al. (2024) show in a comparative analysis of different LLM-based expansion techniques that such expansion tends to yield gains for weaker (retrieval) models, while it can degrade performance for stronger models.

2.4 Fusion Methods

We employ multiple prompt-generated variants of the same document collection for IR and fuse the resulting rankings into a single summarized result. This fusion task falls under *internal metasearch* (also known as *data fusion*), which combines outputs from multiple retrieval systems that share the same document set. As summarized by Montague and Aslam (2002), different retrieval systems perform differently on the same task, and fusing their outputs yields more consistent results.

There exists a multitude of fusion algorithms, both supervised and unsupervised, which differ along several dimensions—most importantly, whether they rely on similarity scores or ranks to form the fused list. Fox and Shaw (1994) propose several score-aggregation methods; Zhang et al. (2002) introduce a method based on reciprocal ranks; and Mourão et al. (2014) extended this idea by considering inverse squared ranks. By contrast, other authors adopt probabilistic approaches that estimate the likelihood of a system returning a relevant document for a given query. In *ProbFuse* (Lillis et al., 2006), estimated probabilities depend on the documents retrieved during training and their positions in the ranking. Other probabilistic methods include *Bayes-Fuse* (Aslam and Montague, 2001), which estimates document relevance by summing, across systems, the log-ratio of the probability that the document at a given rank is relevant versus non-relevant, and *SegFuse* (Shokouhi, 2007), which—similar to *ProbFuse*—partitions rank lists into segments that grow larger down the ranking.

Key	Prompt
EN.v1	*Can you summarize the text keeping the most relevant information and just give me the output without your comments:*
EN.v2	*Can you rewrite the text with other more conspicuous vocabulary and just give me the output without your comments:*
EX.v1	*Can you expand the text with your knowledge keeping, expanding, and explaining the most relevant information and just give me the output without your comments:*
EX.v2	*Can you reduce and summarize the text integrating some of your knowledge, keeping, expanding, and explaining the most relevant information and just give me the output without your comments:*
EL.v1	*Can you rewrite the text with a more negative stance and just give me the output without your comments:*
EL.v2	*Can you expand the text, making comments that should help a person to better understand the content of the text:*

Table 1: Prompts used to generate texts with LLMs. The *Key* indicates the macrocategory and prompt identifier.

Lillis et al. (2010) investigate modeling relevance probabilities without large training sets by using a system's *Mean Average Precision* to relate relevance probability to rank. Additional algorithms arise from *Social Choice Theory*, where retrieval systems act as voters. Boehmer et al. (2023) explain how to use positional scoring rules to transform voters' rankings into scores, while Montague and Aslam (2001) and Montague and Aslam (2002) describe methods based on the *Borda Count Principle* and the *Condorcet* method.

3 Methodology

Our method is based on multiple text expansions via LLMs. In the first step, an LLM is fed each of the n_D documents in the corpus D; for each document, a set of variants is generated using a predefined list of n_P prompts based on the 5E Instructional Model. This yields $n_P + 1$ versions per document (including the original), i.e., $n_D(n_P+1)$ documents in total. We then compute embeddings for documents (original and generated) and queries to measure similarity using *cosine similarity*, a widely used metric. For each query q and each set of generated documents, we build a retrieved set of relevant documents. These retrieved sets are then fused using one or more fusion methods, and the final results are evaluated using nDCG@10.

3.1 Document Expansion via the 5E Model

Although LLMs are often perceived as human-like, they lack genuine human capabilities. Therefore, to employ the 5E model in this study, certain adaptations were necessary. The five phases of the original scheme were used as guidelines for designing prompts to generate alternative text corpora. We make two key modifications: (i) Excluding the *Evaluation* phase, as our goal is not to teach a model, although one could imagine future work where an LLM evaluates and refines its own outputs; and (ii) Merging the *Engage* and *Explore* phases. This leaves three prompting dimensions:

Engage and Explore (EN) The LLM reformulates a given text by altering vocabulary or summarizing it, aiming to mitigate the *Vocabulary Mismatch Problem* while preserving the content.

Explain (EX) The LLM explains the text using its retained knowledge, enriching the content and aligning it more closely with human understanding.

Elaborate (EL) The LLM applies the concepts of a given text to other contexts (e.g., analogies, examples, or tone shifts). Such reformulations may improve alignment between texts and queries.

The exact prompt design (cf. Table 1) inevitably involves some arbitrariness, which we recognize as a key limitation of our work. In preliminary experiments (not reported here), we pre-tested a larger set of prompts based on the 5E Model on a subset of the BEIR dataset. We selected the best performing ones for the main analysis across all NanoBEIR datasets in order to keep computational costs manageable.

4 Experiments

We investigate whether the LLMs' capabilities can be leveraged to summarize or expand a given text, and whether such generated texts can serve as effective substitutes for or complements to the original text in IR tasks. In particular, we aim to combine rankings from multiple generated corpora and

assess whether their fusion improves retrieval performance. We further explore whether adopting the E5 framework to guide text generation yields additional improvements over the original texts and which strategy is most effective. We examine the following hypotheses:

H1 Generated texts will improve performance in IR tasks compared to original texts, and performance will further increase as the number of generated texts used in fusion grows.

H2 Larger LLMs will generate texts that are better suited for IR tasks than smaller models.

H3 Employing a cognitive framework to guide text generation may provide one way to enhance IR performance compared to using the original texts alone.

4.1 Data

Given the computational challenges associated with generating texts, we opted for the NanoBEIR dataset (Thakur et al., 2022), a compact version of BEIR (Thakur et al., 2021), a widely used benchmark in IR research. The NanoBEIR variant contains 50 queries and up to 10k documents per individual dataset, in contrast to the millions of documents included in the standard BEIR datasets. Both comprise generalist and domain-specific datasets, covering a broad range of topics.

4.2 LLMs & Embedding Models

LLMs For text generation, we employ models from the Qwen family (Yang et al., 2024; Qwen Team, 2024) to analyze the impact of model size on retrieval performance. Specifically, we use the medium-sized, instruction-tuned `Qwen2.5` models with 3B, 7B, and 14B parameters, which enables a systematic comparison across different scales.

Embeddings To ensure comparability with state-of-the-art approaches, model selection was guided by the MTEB Leaderboard for retrieval tasks (Muennighoff et al., 2023). We employ the `INF-Retriever-v1-1.5B` (*Infly*, Yang et al., 2025) embeddings for the main analyses (§5), `gte-Qwen2-1.5B-instruct` (Li et al., 2023) for the ablation (§6). While the former is of particular interest due to its results on NanoBEIR, making it especially relevant for evaluating retrieval under resource-constrained conditions, the latter has demonstrated competitive performance across general retrieval benchmarks.

4.3 Fusion Algorithms

In preliminary analyses, we evaluated a range of fusion algorithms commonly discussed in the literature. While retrieval effectiveness was comparable across the best-performing methods[2], execution times varied substantially, particularly for large document corpora. Since design choices in this study are guided by both retrieval effectiveness and computational efficiency, we opted to employ a single method for all analyses: the *logn inverse squared rank* (Mourão et al., 2014). This algorithm provides a favorable trade-off, achieving strong retrieval performance while maintaining manageable computational costs. For the implementation, we relied on the `Ranx` library (Bassani and Romelli, 2022). For score normalization, the widely used *min–max normalization* was applied prior to fusion.

5 Main Results

Table 2 shows average nDCG@10 differences (and standard deviations) between the retrieval results based on only the original corpus and three clusters of strategies: *Single prompts* refers to retrieval using only a single corpus of generated texts, while *Prompt Combinations* and *Prompt Combinations + Original* correspond to fused retrieval results based on multiple generated corpora, with or without the original corpus. Using multiple generated corpora yields average improvements over using only the original corpus in five out of thirteen datasets, while using single generated corpora alone outperforms fusion in only one case (SciFact). When results from generated corpora are fused with retrieval results from the original corpus, gains appear most consistently. Overall, generated texts rarely serve as a full replacement for the original corpus, but when combined with it, they can provide clear improvements—particularly for generalist datasets such as ArguAna, open-domain general QA datasets (NQ and HotpotQA), or scientific/medical datasets (SciFact and NFCorpus).

Table 3 reports how performance differences relative to using only the original corpus vary with the *number* of fused retrieval results based on generated corpora. Although no strictly monotonic trend emerges, the largest average gains—both with and without including the original corpus—tend to

[2]We tested CombMIN, CombMAX, CombMED, CombSUM, CombANZ, CombMNZ (Fox and Shaw, 1994), ISR, log-ISR, logn-ISR (Mourão et al., 2014), RRF (Cormack et al., 2009), MAPFuse (Lillis et al., 2010).

Dataset	arguana	climatefever	dbpedia	fever	fiqa	hotpotqa	msmarco	nfcorpus	nq	quoraretrieval	scidocs	scifact	touche2020
Original (Absolute Values)	63.85	42.64	65.95	93.63	66.82	83.41	64.31	38.70	67.91	95.96	47.67	81.72	52.39
Single Prompts	-0.74 (± 2.58)	-6.28 (± 1.73)	-2.07 (± 1.00)	-1.98 (± 2.26)	-5.45 (± 2.66)	-0.70 (± 1.56)	-7.43 (± 3.11)	-0.08 (± 0.75)	-1.27 (± 3.05)	-9.30 (± 2.98)	-2.90 (± 1.29)	0.83 (± 1.61)	-3.78 (± 1.36)
Prompt Combinations	1.38 (± 1.53)	-5.95 (± 1.28)	-1.36 (± 1.05)	-1.60 (± 1.23)	-3.18 (± 1.23)	0.34 (± 0.73)	-5.92 (± 1.83)	0.24 (± 0.58)	0.05 (± 2.32)	-5.62 (± 2.42)	-2.35 (± 1.19)	1.12 (± 1.34)	-2.58 (± 0.67)
Prompt Combinations with Original	1.82 (± 0.95)	-4.31 (± 1.06)	-1.05 (± 0.75)	-0.42 (± 0.47)	-1.35 (± 0.67)	0.81 (± 0.54)	-3.72 (± 1.39)	0.39 (± 0.48)	0.74 (± 1.41)	-2.12 (± 1.44)	-1.49 (± 1.01)	1.09 (± 0.92)	-1.49 (± 0.41)
Mean difference to Original	0.82 (± 1.12)	-5.51 (± 0.86)	-1.49 (± 0.43)	-1.33 (± 0.66)	-3.33 (± 1.68)	0.15 (± 0.63)	-5.69 (± 1.52)	0.19 (± 0.20)	-0.16 (± 0.83)	-5.68 (± 2.93)	-2.25 (± 0.58)	1.02 (± 0.13)	-2.62 (± 0.94)

Table 2: Averaged nDCG@10 differences (± std. dev.) across all models and prompts for the different datasets and prompt combinations. Green indicates improvements, red a decrease, relative to the Original baseline.

nr_fused_texts	arguana	climatefever	dbpedia	fever	fiqa	hotpotqa	msmarco	nfcorpus	nq	quoraretrieval	scidocs	scifact	touche2020	Mean
0 (only Original)	63.85	42.64	65.95	93.63	66.82	83.41	64.31	38.70	67.91	95.96	47.67	81.72	52.39	66.54
1	-0.74 (± 2.58)	-6.28 (± 1.73)	-2.07 (± 1.00)	-1.98 (± 2.26)	-5.45 (± 2.66)	-0.70 (± 1.56)	-7.43 (± 3.11)	-0.08 (± 0.75)	-1.27 (± 3.05)	-9.30 (± 2.98)	-2.90 (± 1.29)	0.83 (± 1.61)	-3.78 (± 1.36)	-3.17
2	0.40 (± 1.54)	-5.94 (± 1.23)	-1.48 (± 1.24)	-2.19 (± 1.33)	-4.25 (± 1.10)	-0.03 (± 0.82)	-6.41 (± 2.10)	0.05 (± 0.72)	-0.47 (± 2.68)	-7.55 (± 2.20)	-2.65 (± 1.29)	0.88 (± 1.44)	-2.84 (± 0.68)	-2.50
2+Original	1.43 (± 0.88)	-3.54 (± 0.71)	-0.93 (± 0.89)	-0.37 (± 0.33)	-1.33 (± 0.91)	0.68 (± 0.68)	-3.15 (± 1.39)	0.15 (± 0.54)	0.49 (± 1.30)	-2.43 (± 1.50)	-1.16 (± 0.91)	0.95 (± 1.00)	-1.38 (± 0.54)	-0.81
4	2.11 (± 1.11)	-5.97 (± 1.38)	-1.24 (± 1.08)	-1.26 (± 0.94)	-2.45 (± 0.49)	0.59 (± 0.60)	-5.54 (± 1.52)	0.40 (± 0.41)	0.55 (± 2.13)	-4.37 (± 1.46)	-2.04 (± 1.12)	1.14 (± 1.28)	-2.38 (± 0.72)	-1.57
4+Original	2.07 (± 0.81)	-4.74 (± 0.85)	-1.14 (± 0.75)	-0.43 (± 0.54)	-1.39 (± 0.47)	0.86 (± 0.41)	-3.99 (± 1.06)	0.56 (± 0.40)	1.09 (± 1.42)	-1.75 (± 1.55)	-1.75 (± 1.04)	1.13 (± 0.86)	-1.51 (± 0.29)	-0.85
6	2.14 (± 1.16)	-5.90 (± 1.61)	-1.34 (± 0.32)	-0.82 (± 1.23)	-2.18 (± 0.55)	0.70 (± 0.40)	-5.60 (± 2.17)	0.32 (± 0.60)	0.11 (± 2.12)	-3.59 (± 0.61)	-2.36 (± 1.29)	1.79 (± 1.47)	-2.43 (± 0.30)	-1.47
6+Original	2.25 (± 1.43)	-5.32 (± 1.15)	-1.15 (± 0.40)	-0.54 (± 0.71)	-1.25 (± 0.53)	1.01 (± 0.52)	-4.64 (± 2.02)	0.61 (± 0.29)	0.43 (± 2.01)	-2.34 (± 1.00)	-1.70 (± 1.29)	1.44 (± 1.10)	-1.75 (± 0.24)	-1.00
Mean difference to Original	1.38 (± 1.06)	-5.38 (± 0.89)	-1.34 (± 0.34)	-1.08 (± 0.69)	-2.62 (± 1.51)	0.44 (± 0.56)	-5.25 (± 1.35)	0.29 (± 0.24)	0.13 (± 0.72)	-4.47 (± 2.67)	-2.08 (± 0.56)	1.17 (± 0.32)	-2.30 (± 0.78)	-1.62

Table 3: Averaged nDCG@10 differences (± std. dev.) across all models and prompts for the different datasets and numbers of combined prompts, with Infly embeddings. Green indicates improvements, red a decrease, relative to the Original baseline.

occur when more texts are fused, with the greatest improvements appearing when retrieval results from the original corpus are fused together with those from generated texts. This suggests that increasing the number of fused texts can help the system converge on a shared notion of which documents are relevant, illustrating the *Chorus Effect* in data fusion (Vogt and Cottrell, 1999). Figure 2 (Appendix A) provides a graphical representation of Table 3, showing the average differences and the corresponding standard deviations. For most datasets, the substantial variability in performance suggests that improvements are driven not only by the act of fusing retrieval results from different generated texts, but also by which specific texts are combined. These results provide partial support for H1, as using generated texts alone generally does not outperform the original corpus. However, when used in conjunction with it, they often enhance retrieval performance.

Table 4 reports the average performance across fusion strategies relative to the original corpus and provides evidence to investigate H2. The results indicate that larger LLMs do not yield consistently greater improvements. As we focus on using LLMs for expansion rather than on retrieval models themselves, this observation is broadly consistent with Weller et al. (2024), who report that expansion can harm the performance of stronger retrieval models. However, gains are observed in datasets previously found to benefit most from fusion strategies—namely, open-domain question answering

LLM	arguana	climatefever	dbpedia	fever	fiqa	hotpotqa	msmarco	nfcorpus	nq	quoraretrieval	scidocs	scifact	touche2020
Original	63.85	42.64	65.95	93.63	66.82	83.41	64.31	38.70	67.91	95.96	47.67	81.72	52.39
Qwen2.5-3B-Instruct	2.21 (± 1.29)	-4.88 (± 1.13)	-1.07 (± 0.95)	-1.63 (± 0.78)	-2.95 (± 2.07)	0.18 (± 0.72)	-5.84 (± 2.36)	0.40 (± 0.55)	-0.14 (± 1.21)	-5.61 (± 3.42)	-0.97 (± 0.91)	0.03 (± 1.06)	-2.05 (± 1.03)
Qwen2.5-7B-Instruct	0.53 (± 2.19)	-6.64 (± 1.61)	-2.06 (± 0.71)	-0.96 (± 2.28)	-3.50 (± 2.54)	0.08 (± 1.60)	-4.56 (± 2.65)	0.23 (± 0.78)	-1.18 (± 2.91)	-4.81 (± 4.11)	-2.71 (± 1.01)	1.23 (± 1.14)	-2.73 (± 1.17)
Qwen2.5-14B-Instruct	-0.06 (± 1.93)	-4.91 (± 1.39)	-1.26 (± 1.10)	-1.31 (± 1.31)	-3.21 (± 2.47)	0.32 (± 1.08)	-6.41 (± 2.62)	-0.03 (± 0.46)	1.01 (± 2.39)	-6.08 (± 3.62)	-2.96 (± 0.88)	1.82 (± 0.99)	-2.90 (± 1.47)
Mean difference to Original	0.89 (± 0.96)	-5.48 (± 0.82)	-1.46 (± 0.43)	-1.30 (± 0.27)	-3.22 (± 0.22)	0.19 (± 0.10)	-5.60 (± 0.77)	0.20 (± 0.18)	-0.10 (± 0.89)	-5.50 (± 0.52)	-2.21 (± 0.89)	1.03 (± 0.74)	-2.56 (± 0.37)

Table 4: Averaged nDCG@10 differences (± std. dev.) across all different (numbers of) prompts for the different datasets and for different LLMs from the Qwen2.5 family and Infly embeddings. Green indicates improvements, red a decrease, relative to the Original baseline.

Strategy	arguana	climatefever	dbpedia	fever	fiqa	hotpotqa	msmarco	nfcorpus	nq	quoraretrieval	scidocs	scifact	touche2020	Sum
Original	0	3	1	2	3	0	3	0	0	2	2	0	3	19
Original+Combination	2	0	1	1	0	1	0	1	1	1	1	0	0	9
EN+Combination+Original	2	0	0	1	0	0	0	1	1	1	0	0	0	6
EX+Combination+Original	1	0	1	1	0	0	0	0	1	0	1	0	0	5
EL+Combination+Original	1	0	0	0	0	1	0	1	0	1	0	0	0	4
EX+Combination	0	0	0	0	0	1	0	1	2	0	0	0	0	4
EN+Combination	1	0	1	0	0	2	0	0	0	0	0	0	0	4
EL+Combination	1	0	1	0	0	0	0	0	2	0	0	0	0	4
EL	0	0	0	0	0	0	0	0	0	0	0	2	0	2
EN	0	0	0	0	0	0	0	0	0	0	0	1	0	1
EX	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Table 5: Frequencies of best-performing combinations across datasets and LLMs for different prompt strategies and their combinations. Infly embeddings are used. Each cell indicates the number of times a given strategy achieved the best performance, with the final column showing totals across datasets. When a column sum exceeds the number of LLMs (3), it indicates that multiple prompt groups were part of the combination yielding the best performance.

(NQ and HotpotQA). This outcome partially supports H2, suggesting that scaling model size can improve performance in open-domain tasks; however, it does not guarantee consistent benefits across all tasks.

Table 5 examines the effect of different prompt types, derived from the 5E cognitive model, on retrieval performance. The table shows how often a prompt (combination) achieved the highest absolute retrieval performance across all LLMs considered. The original corpus outperformed in most cases; nevertheless, combining it with generated texts led to non-negligible improvements in many instances. When comparing the three prompting categories, EN and EX prompts consistently performed better, in combination with or without the original corpus, than EL prompts. EN prompts focus on summarizing the original text, while EX prompts aim to expand it with knowledge retained by the LLM, and EL prompts seek to re-elaborate the original text (e.g., reformulating it with a more negative stance). These results suggest that summarizing or expanding texts with LLMs can enhance retrieval performance, particularly when fused with the original corpus, whereas altering the text (EL) may not always yield comparable improvements. These findings suggest that employing a structured framework—such as the cognitively motivated one used in this work—can help more effectively leverage LLMs' retained knowledge, yielding expansions that translate into improvements in retrieval performance. Thus, these observations support H3.

6 Ablations

Given the multitude of factors influencing retrieval performance, we conduct ablations regarding the two most important ones: the LLM for generation and the embedding model for retrieval.

Dataset	arguana	climatefever	dbpedia	fever	fiqa	hotpotqa	msmarco	nfcorpus	nq	quoraretrieval	scidocs	scifact	touche2020
Original	63.85	42.64	65.95	93.63	66.82	83.41	64.31	38.70	67.91	95.96	47.67	81.72	52.39
Single Prompts (7B)	-1.61 (± 2.95)	-7.45 (± 1.56)	-2.70 (± 0.46)	-2.12 (± 3.77)	-6.33 (± 1.94)	-0.93 (± 2.57)	-6.74 (± 3.35)	-0.34 (± 0.96)	-2.87 (± 4.15)	-8.67 (± 4.18)	-3.47 (± 1.05)	1.12 (± 1.34)	-4.06 (± 0.80)
Single Prompts (14B)	-1.76 (± 2.41)	-5.68 (± 2.09)	-1.72 (± 1.28)	-2.06 (± 1.63)	-5.64 (± 3.04)	-0.78 (± 0.92)	-8.40 (± 3.21)	-0.18 (± 0.46)	-0.37 (± 2.78)	-9.90 (± 2.17)	-3.62 (± 0.54)	1.45 (± 1.45)	-4.22 (± 1.87)
Prompt Combinations (7B)	1.11 (± 0.87)	-7.44 (± 0.90)	-2.18 (± 0.52)	-0.91 (± 1.51)	-3.40 (± 1.66)	0.32 (± 0.87)	-4.52 (± 1.88)	0.26 (± 0.64)	-1.07 (± 2.58)	-5.07 (± 2.67)	-2.85 (± 0.69)	1.23 (± 1.24)	-2.74 (± 0.67)
Prompt Combinations (14B)	0.22 (± 1.32)	-5.15 (± 0.54)	-1.09 (± 1.18)	-1.64 (± 1.15)	-2.92 (± 1.18)	0.52 (± 0.84)	-6.67 (± 1.38)	0.03 (± 0.54)	1.41 (± 2.55)	-6.37 (± 2.46)	-3.26 (± 0.65)	2.29 (± 0.71)	-3.01 (± 0.47)
Prompt Combinations with Original (7B)	1.80 (± 0.62)	-5.14 (± 1.18)	-1.41 (± 0.48)	-0.01 (± 0.13)	-1.17 (± 0.46)	0.70 (± 0.54)	-2.73 (± 0.88)	0.69 (± 0.44)	0.17 (± 0.86)	-1.24 (± 1.39)	-1.93 (± 0.74)	1.32 (± 1.03)	-1.59 (± 0.28)
Prompt Combinations with Original (14B)	1.13 (± 0.72)	-3.99 (± 0.72)	-1.05 (± 0.89)	-0.34 (± 0.27)	-1.42 (± 0.70)	1.05 (± 0.61)	-4.44 (± 1.66)	0.05 (± 0.42)	1.79 (± 1.54)	-2.53 (± 1.44)	-2.08 (± 0.59)	1.66 (± 0.64)	-1.65 (± 0.30)
Mean difference to Original	0.15 (± 1.51)	-5.81 (± 1.38)	-1.69 (± 0.65)	-1.18 (± 0.90)	-3.48 (± 2.13)	0.15 (± 0.81)	-5.58 (± 2.05)	0.09 (± 0.36)	-0.16 (± 1.71)	-5.63 (± 3.38)	-2.87 (± 0.72)	1.51 (± 0.42)	-2.88 (± 1.13)

Table 6: Averaged nDCG@10 differences (± std. dev.) across all prompts for the different datasets and prompt combinations for the two largest of the three LLMs, `Qwen2.5-7B-Instruct` and `Qwen2.5-14B-Instruct`, with Infly embeddings. Green indicates improvements, red a decrease, relative to the Original baseline.

Dataset	arguana	climatefever	dbpedia	fever	fiqa	hotpotqa	msmarco	nfcorpus	nq	quoraretrieval	scidocs	scifact	touche2020
Original (Absolute Values)	28.96	25.65	39.67	82.91	15.24	70.40	21.99	15.52	16.40	93.47	26.79	57.04	37.95
Single Prompts	-8.14 (± 5.54)	-5.89 (± 2.17)	0.98 (± 2.91)	-10.74 (± 7.03)	16.74 (± 3.54)	-6.91 (± 5.08)	4.05 (± 3.65)	4.99 (± 2.17)	7.80 (± 6.97)	-32.48 (± 9.22)	-8.39 (± 4.77)	4.37 (± 5.01)	-4.54 (± 3.40)
Prompt Combinations	-5.45 (± 3.31)	-4.57 (± 1.63)	4.23 (± 1.54)	-6.12 (± 3.40)	21.22 (± 2.65)	-2.94 (± 2.49)	7.75 (± 2.52)	6.84 (± 1.54)	12.18 (± 6.44)	-20.81 (± 7.56)	-5.83 (± 3.03)	7.37 (± 3.54)	-1.22 (± 1.97)
Prompt Combinations with Original	-1.70 (± 3.18)	-3.38 (± 1.04)	3.95 (± 1.38)	-2.48 (± 1.85)	19.41 (± 2.94)	-0.33 (± 0.92)	6.89 (± 1.96)	5.34 (± 1.43)	9.49 (± 6.15)	-7.90 (± 3.35)	-1.53 (± 2.24)	6.98 (± 3.21)	1.01 (± 1.69)
Mean difference to Original	-5.10 (± 2.64)	-4.61 (± 1.03)	3.05 (± 1.47)	-6.45 (± 3.38)	19.13 (± 1.84)	-3.40 (± 2.70)	6.23 (± 1.58)	5.72 (± 0.80)	9.82 (± 1.80)	-20.39 (± 10.04)	-5.25 (± 2.83)	6.24 (± 1.33)	-1.58 (± 2.28)

Table 7: Averaged nDCG@10 differences (± std. dev.) across all prompts for the different datasets and prompt combinations, with semantic similarity computed using `gte-Qwen2-1.5B-instruct` embeddings. Green indicates improvements, red a decrease, relative to the Original baseline.

LLMs The following tables reproduce Table 2 but only considering generation with the medium-sized LLM and the larger LLM (`Qwen2.5-7B-Instruct` and `Qwen2.5-14B-Instruct`, Table 6). As tables show, there is no *consistent* improvement over all datasets when using only a larger LLM. When using combinations of prompts, both models surpass the other on 50% of the datasets; when including the original as well, the *smaller* of the two models performs favorably in 7 out of 13 cases.

Embedding Model Table 7 reproduces table 2 but with `gte-Qwen2-1.5B-instruct` embeddings. With this embedding model, the baseline nDCG@10 is lower than with the Infly embeddings, and expansion appears to particularly benefit weaker embedding models. In the main results and in preliminary (unreported) experiments with E5 (Wang et al., 2022), BGE (Xiao et al., 2023), and UAE (Li and Li, 2023) embedding models, fusion that included the original corpus consistently performed best, as one would expect. Interestingly, with this embedding model, fusion over generated corpora without the original corpus slightly outperforms fusion that includes it.

7 Conclusion and Future Work

The goal of this study was to investigate whether LLMs, guided by a set of cognitively motivated prompts, can be used to generate alternative versions of a given text corpus that enhance performance in retrieval tasks. Our results show that while using *single* generated texts underperforms

compared to using the original corpus, retrieval results obtained using combinations of the generated alternative corpora can improve performance. The effect is even more apparent when they are used in combination with the original corpus. Moreover, using prompts that expand texts by leveraging the retained knowledge of LLMs to complement and summarise the corpus (EN and EX prompts) yields greater improvements in retrieval performance than more exotic reformulations of the text (EL prompt), highlighting that LLMs can act as powerful allies in retrieval tasks.

However, this study and its findings raise several questions that should be addressed in future work. First, our hypotheses should be validated on additional datasets, ideally larger than those considered here. Second, alternative prompting schemes could be explored: rather than applying a single set of prompts to a heterogeneous group of datasets, prompting strategies might be tailored to the specific domain of each dataset. Third, although using the 5E Model to guide prompt design yielded relevant results, it represents only one possible approach to developing an evidence-based prompting strategy, rather than the only one. Future work could therefore investigate alternative prompting frameworks. Finally, a natural next step would be to investigate whether these cognitively motivated expansions can also improve the performance of dense retrievers.

Limitations

This study has several limitations. First, we rely on LLMs to generate texts from human-crafted prompts. Although the 5E framework guided prompt design, prompt writing remains inherently subjective, and the results and conclusions reported here may not hold under alternative prompting strategies. Moreover, the same fixed set of prompts is applied to text corpora that differ considerably from one another. Second, limited computational resources prevented us from systematically evaluating larger LLMs or using larger datasets. Our approach is feasable with smaller document corpora but pose significant computuational challenges for larger text collections.[3] Third, our analysis is exploratory in nature. For example, when comparing performance differences between generated texts and the original corpus, we did not conduct statistical significance tests to assess the robustness of these differences. Finally, when reproducing retrieval results from the MTEB leaderboard (Muennighoff et al., 2023)—without using any LLMs—the results reported in this study do not exactly match the leaderboard scores, although the deviations remain within an acceptable range. Several factors may have contributed to these discrepancies, including variations in the instructions used to supplement the input during embedding computation.

[3] As an example, in preliminary analyses, using an NVIDIA Tesla V100 GPU with 16GB of memory, for the (full) FiQa dataset, which includes more than 57K documents, over 600 queries, and six prompts for text generation, the total runtime was 44.35 hours. The majority of this time, 30.93 hours (70%), was dedicated to text generation using the six prompts. Computing the embeddings for documents and queries took 1.39 hours (3%) while calculating similarity scores required 11.35 hours (26%). Finally, the fusion and evaluation process took 0.68 hours (2%).

Acknowledgments

Matthias Aßenmacher received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure – NFDI 27/1 - 460037581 - BERD@NFDI. We acknowledge EuroHPC JU for awarding the project ID EHPC-DEV-2024D05-033 access to the Meluxina Computing Services.

References

Javed A Aslam and Mark Montague. 2001. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284.

Elias Bassani and Luca Romelli. 2022. ranx.fuse: A python library for metasearch. In *CIKM*, pages 4808–4812. ACM.

Niclas Boehmer, Robert Bredereck, and Dominik Peters. 2023. Rank aggregation using scoring rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5515–5523.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.

Rodger W Bybee. 1990. Science for life & living: An elementary school science program from biological sciences curriculum study. *The American Biology Teacher*, 52(2):92–98.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Lena Ballone Duran and Emilio Duran. 2004. The 5e instructional model: A learning cycle approach for inquiry-based science teaching. *Science Education Review*, 3(2):49–58.

Miles Efron, Peter Organisciak, and Katrina Fenlon. 2012. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 911–920.

Edward Fox and Joseph Shaw. 1994. Combination of multiple searches. *NIST special publication SP*, pages 243–243.

George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Soyeong Jeong, Jinheon Baek, ChaeHun Park, and Jong C Park. 2021. Unsupervised document expansion for information retrieval with stochastic text generation. *arXiv preprint arXiv:2105.00666*.

Candace Joswick and Melissa Hulings. 2024. A systematic review of bscs 5e instructional model evidence. *International Journal of Science and Mathematics Education*, 22(1):167–188.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

David Lillis, Fergus Toolan, Rem Collier, and John Dunnion. 2006. Probfuse: a probabilistic approach to data fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 139–146.

David Lillis, Lusheng Zhang, Fergus Toolan, Rem W Collier, David Leonard, and John Dunnion. 2010. Estimating probabilities for effective data fusion. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 347–354.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mark Montague and Javed A Aslam. 2001. Relevance score normalization for metasearch. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 427–433.

Mark Montague and Javed A Aslam. 2002. Condorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 538–548.

André Mourão, Flávio Martins, and Joao Magalhaes. 2014. Inverse square rank fusion for multimodal search. In *2014 12th international workshop on content-based multimedia indexing (CBMI)*, pages 1–6. IEEE.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019a. From doc2query to doctttttquery. *Online preprint*, 6(2).

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019b. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Héctor Ruiz-Martín and Rodger W Bybee. 2022. The cognitive principles of learning underlying the 5e model of instruction. *International Journal of STEM Education*, 9(1):21.

Milad Shokouhi. 2007. Segmentation of search engine results for effective data-fusion. In *European Conference on Information Retrieval*, pages 185–197. Springer.

Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 407–414.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2022. Nanobeir: A lightweight subset of the beir benchmark. https://huggingface.co/datasets/BeIR/nano-beir. Subset of the BEIR benchmark for efficient evaluation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Christopher C Vogt and Garrison W Cottrell. 1999. Fusion via a linear combination of scores. *Information retrieval*, 1(3):151–173.

Ellen M Voorhees. 1994. Query expansion using lexical-semantic relations. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 61–69. Springer.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. 2024. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1987–2003.

Yu Xia, Junda Wu, Sungchul Kim, Tong Yu, Ryan A Rossi, Haoliang Wang, and Julian McAuley. 2025. Knowledge-aware query expansion with large language models for textual and relational retrieval. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4275–4286.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Junhan Yang, Jiahe Wan, Yichen Yao, Wei Chu, Yinghui Xu, and Yuan Qi. 2025. inf-retriever-v1 (revision 5f469d7).

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Min Zhang, Ruihua Song, Chuan Lin, Shaoping Ma, Zhe Jiang, Yijiang Jin, Yiqun Liu, and Le Zhao. 2002. Thu trec 2002: Novelty track experiments. In *TREC*.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *ACM Transactions on Information Systems*.

A Performance for different number of prompts

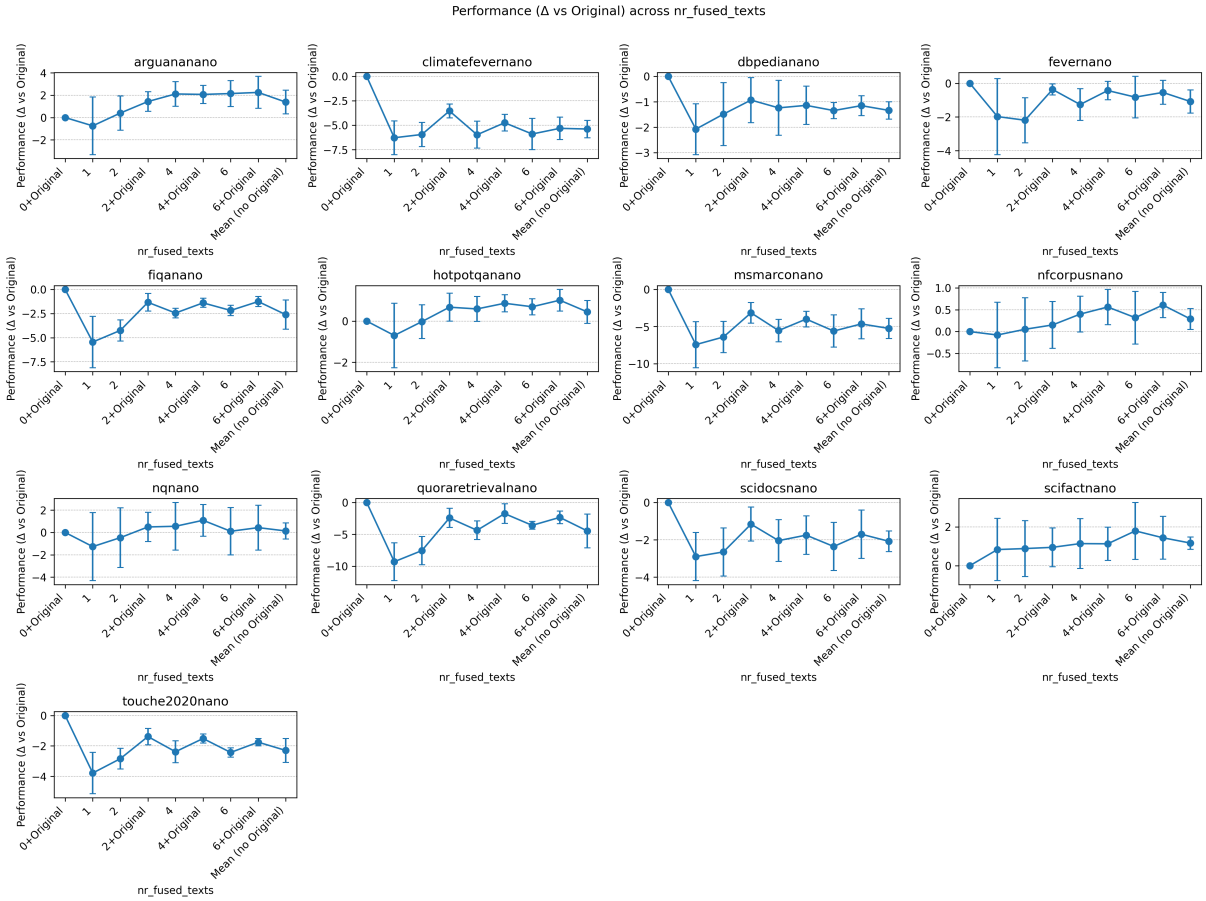

Figure 2: Averaged nDCG@10 differences (± std. dev.) across all models and prompts for the different datasets and prompt combinations, with Infly embeddings.