Enhancing RAG's Retrieval via Query Backtranslations

Delaram Rajaei, Zahra Taheri, and Hossein Fani(✉)

University of Windsor, Windsor, ON, Canada
{rajaeid,taherik,hfani}@uwindsor.ca

Abstract. Retrieval-augmented generation (rag) systems extend the capabilities of generating responses beyond the pretrained knowledge of large language models by augmenting the input prompt with relevant documents retrieved by an information retrieval system, which is of particular importance when knowledge is constantly updated and cannot be memorized by the model. Rag-based systems operate in two phases: retrieval and generation. In the retrieval phase, documents are retrieved from various versions of the original query, then fused and reranked to create a unified list, and the more relevant list of documents, the better the subsequent generation phase. In this paper, we propose an unsupervised method to enhance the retrieval phase by transforming an original query into newly reformulated versions without semantic drift to enhance the relevance of the retrieved documents. Specifically, for an original query, (1) we generate its backtranslated versions via different languages, (2) retrieve an ordered list of relevant documents for each backtranslated version, and finally, (3) merge the lists of retrieved documents into a single ranked list via reciprocal rank fusion. Our extensive experiments across 5 query sets with different query topics and 10 languages from 7 language families using 2 neural machine translators demonstrated the effectiveness of our proposed method in enhancing rag's retrieval in comparison with existing unsupervised query expanders. We open-sourced our research at https://github.com/fani-lab/RePair/tree/rrf-wise24.

Keywords: Query Reformulation · Backtranslation · Reciprocal Rank Fusion · Retrieval-Augmented Generation

1 Introduction

Retrieval-augmented generation (rag) integrates external information retrieval mechanisms to enhance large language models (llms) using two phases: (1) retrieval phase, where it retrieves relevant documents and information related to the original query, and (2) generation phase, where the retrieved documents, combined with the original query, are provided as input to the language model to generate a response. Rag systems have found applications in diverse fields, including product search in e-commerce [47], data-to-text generation [17], and enhancing document retrieval and robustness [66].

M. Barhamgi et al. (Eds.): WISE 2024, LNCS 15436, pp. 270–285, 2025.
https://doi.org/10.1007/978-981-96-0579-8_20

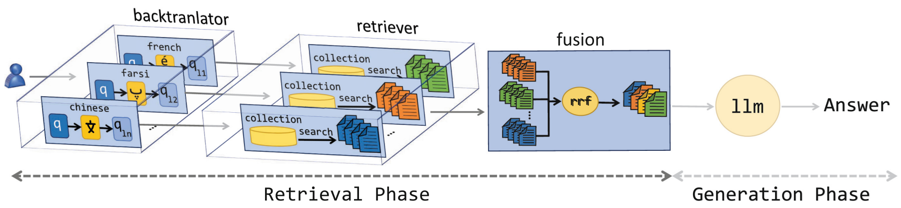

Fig. 1. Generating backtranslated versions of a query and fusing retrieved documents for rag-based retrieval.

Traditionally, rag systems rely on the input query to retrieve relevant documents from commercial search engines or neural rankers trained on external knowledge sources [33]. While effective, this approach may yield limited contextual understanding of queries and suboptimal retrieval outcomes, particularly in addressing complex or diverse information needs [50]. Rag-fusion advances traditional rag systems by generating reformulations of the original query [47] and merging the retrieved documents per query variation into a single unified list. This approach enriches the retrieval context, allowing for more comprehensive responses by considering diverse perspectives of user queries [20,58]. State-of-the-art query reformulations are largely based on fine-tuning transformers. Arabzadeh et al. [4] and others [42] proposed fine-tuning `t5` to generate reformulated queries. Zheng et al. [67] leveraged `bert` to mitigate non-relevant information in query reformulation. Fine-tuning transformers, however, demands computational resources and has environmental impacts [56]. Moreover, their efficacy is questionable since evaluation data may have been seen during pretraining, risking data leakage and overestimating their capabilities [28].

In this paper, we propose query backtranslations, that is, translating an original query into other languages and back to the original language, to generate diverse yet contextually relevant query reformulations [53]. Our proposed query backtranslation is a novel yet simple *unsupervised* approach, which maintains relevance and controls topic drift. Figure 1 illustrates the generation of backtranslated versions followed by the fusion of retrieved document sets into a unified set for the generation phase.

2 Related Work

The work related to this paper can be broadly categorized into (1) rag-fusion, (2) query reformulation methods, and (3) natural language backtranslation.

2.1 RAG Fusion

Retrieval-augmented generation (rag) has emerged as a promising avenue for enhancing the performance of large language models (llms). Llms are prone to generating unreliable responses, often influenced by outdated or incorrect information, leading to concerns about hallucinations during content generation [49]. Addressing the limitations of llms has inspired the exploration of retrieval-augmented models aimed at improving response accuracy and reliability [50] through two phases: retrieval and generation. In this paper, we concentrate on the retrieval phase, as improving the accuracy of the document list has been shown to enhance the subsequent generation stage [22,33,41]. The retrieval component is commonly implemented either by using a prebuilt retrieval model or developing a custom retrieval model. Regarding the prebuilt retrieval model, researchers typically utilize one of the following approaches: (1) commercial search engines [41], (2) neural ranking models trained on relevance annotations on external knowledge sources [33], and (3) term matching retrieval models like `bm25` [22]. The application of rag spans various fields such as healthcare [31], cybersecurity [16], legislative document drafting [8], social media [15], and data-to-text generation [17]. Additionally, in enhancing llm robustness, Yan et al. [66] introduce `crag`, which improves the quality of document retrieval and selectively focuses on key information to boost performance in both short- and long-form generation tasks. Such studies underscore rag's versatility and potential to enhance information retrieval and response generation across various fields.

Traditional rag relies on a single query for document retrieval. In contrast, rag-fusion uses multiple queries derived from the original query to enhance the context for information retrieval and includes an extra ranking step that enhances queries by incorporating potentially relevant information from a wider context [13,47]. Rackauckas et al. [47] examine rag-fusion to improve chatbots, demonstrating enhanced accuracy and relevance in responses to customer queries. Diaz et al. [13] highlight the efficiency of using expansion terms from pseudo-relevance feedback in reranking. Most of these studies primarily concentrated on utilizing large language models for questions and answering tasks. In the landscape of query reformulation, traditional methods are being fused by advanced methods such as reciprocal rank fusion (`rrf`), a technique that combines outcomes from different strategies to refine rankings [13,47]. Operating as an extra ranking step, `rrf` prioritizes documents based on their relative ranks across different search methods, resulting in a coherent list of prioritized documents.

2.2 Query Reformulation

Earlier methods in query reformulation mainly relied on unsupervised techniques, utilizing external sources like thesauri or inter-term correlations for modifying queries [37,57]. Feedback from users, either through relevance feedback or pseudo-relevance feedback, was introduced to mitigate issues of semantic drift [52,65]. To address limitations with short queries, semi-supervised and supervised techniques learned user intent from search logs, providing reformulated queries based on semantic and contextual aspects [2,12,59]. Diverse strategies such as hierarchical recurrent encoder-decoder and seq-to-seq models with term-level attention were proposed for effective query reformulation [2,12,59].

Recent efforts focus on creating standard benchmark datasets to train and evaluate supervised query reformulation methods, addressing issues related to semantic drift in both web and non-web information retrieval systems [4,48,60]. The majority of these researchers conducted their studies on well-known `trec` datasets. By developing these benchmarks, the research community aims to establish a common ground for comparing the effectiveness of different query reformulation techniques and ensuring consistent improvements in information retrieval systems. These advancements collectively contribute to more robust and contextually aware retrieval processes, ultimately enhancing user satisfaction and the overall effectiveness of search engines. To the best of our knowledge, no one has yet explored the synergistic impact of backtranslation as a query reformulation method except that of Rajaei et al. [48], which was studied for adhoc web search systems. We are the first to study the impact of backtranslation on the retrieved documents for rag-based systems.

2.3 Natural Language Backtranslation

Natural languages, as primary tools of communication, enable the exchange of thoughts and convey the culture, history, and heritage of a community [23]. Despite shared linguistic universals rooted in the common neurobiological basis of the human brain [19], languages exhibit surface-level differences in structure and semantics, contributing to conveying pragmatics. Backtranslation leverages these disparities, creating new sentence versions through a round-trip translation from a source language to a target language and back (backward translation). Backtranslation yields a new version of the sentence with different and diverse wordings while the meaning remains intact, and hence, has found immediate applications for a wide range of natural language processing tasks as a (1) data augmentation technique such as in machine translation [14,34], document classification [29], review analysis [27], and question-answering [5], or (2) as a quality estimator in evaluating the quality of translations without human-translated references [40,69]. As an augmentation technique, Li et al. [34] and Haq et al. [24] employed backtranslation to generate synthetic parallel corpora in low-resource languages and to scale up the training set for neural machine translators. Ibrahim et al. [29] tackled the class imbalance in training sets for online offensive content detection. Hemmatizadeh et al. [27] tapped into backtranslation to empower the aspect-based sentiment classifiers and detect *latent* aspects. Bhaisaheb et al. [5] iteratively augmented a set of reasoning questions about data charts to leverage *compositional generalization*, i.e., producing *unseen* meaningful combinations of seen terms in sentences, and to improve generating analytical answers via sql programs using `codet5` [64]. For quality estimation, Moon et al. [40] and others [69] use backtranslation as a semantic-level metric for multilingual two-way machine translation when no human-translated reference is available. The approach mimics end-users who assess the quality of an online multilingual translator by comparing the original sentence in a source language and the backtranslated sentence via a target language that they do not understand. Backtranslation as a quality metric outweighs reference-based metrics such as `blue`, which are limited to surface-level lexical similarity. Nonetheless, while backtranslation has been widely used in nlp, its effectiveness for rag-based systems has remained unclear, and we are the first to investigate it.

3 Problem Definition

Given an original query q, its *retrieved* ranked list of relevant documents $\mathcal{D}_q$ by a retriever r, and its *true* list of relevant documents (relevant judgment) $\mathcal{J}_q$, our task is to generate n different versions of q, denoted by $\mathcal{Q} = \{q_i\}_{i=1}^{n}$, using query backtranslation, each with its own retrieved ranked list of relevant documents D_{q_i} by the retriever r, such that the reciprocal rank fusion (`rrf`) [11] of n ranked lists of $\mathcal{D}_{q_i}; 1 \leq i \leq n$, denoted by $\mathcal{D}_q^*$ has a better relevance for a rag-based system in terms of an evaluation metric m.

4 Proposed Approach

To generate the variations of an original query, we propose natural language backtranslation. Let $\mathcal{L}$ be a set of languages. Given a query q, we backtranslate the query, resulting in a set of modified queries $q_{\mathcal{L}} = \{q_l : \forall l \in \mathcal{L}\}$. In our study, without loss of generality to any machine translation models, we leverage meta's *'no language left behind'* (`nllb`)[1], an open-source neural machine translator capable of providing high-quality translations directly between 200 languages [62]. We opt for `nllb` for its particular focus on realizing a *universal* translator while prioritizing low-resource natural languages, as opposed to a small dominant subset of natural languages; it enables query backtranslation augmentation via a vast variety of natural languages with distinct properties. Further, `nllb` is open-sourced to foster transparency and can be smoothly integrated into any pipeline with few lines of code.

Given the retrieved documents $\mathcal{D}$ per backtranslated query q_l using the information retrieval method r, denoted by $\mathcal{D}_{q_l}$, we apply reciprocal rank fusion (`rrf`) [11] to merge $\forall q_l \in q_{\mathcal{L}}; \mathcal{D}_{q_l}$ into a new *single* ranked list $\mathcal{D}_q^*$ based on:

$$\texttt{rrf}(d \in \mathcal{D}_q^*) = \sum_{\mathcal{D}_{q_l} \in q_L} \frac{1}{k + rank(d)} \tag{1}$$

where $rank(d)$ represents the rank of document d in the list retrieved documents $\mathcal{D}_{q_l}$ for the backtranslated query q_l and the constant k mitigates the impact of excessively high rankings as outliers. A higher k value diminishes the influence of higher rankings, thereby ensuring that the final rankings are less skewed by outliers. Afterward, the final list $\mathcal{D}_q^*$ is evaluated using an information retrieval metric m for the query q, denoted as $m(\mathcal{D}_q^*; \mathcal{J}_q)$. We select reciprocal rank fusion because while highly ranked documents hold greater significance, the importance of lower-ranked ones should also be regarded.

[1] github.com/facebookresearch/fairseq/tree/nllb.

Table 1. Statistics on query sets, query length $|q|$, and relevance judgments $\mathcal{J}$.

						avg $m_r(q, \mathcal{J}_q)$			
						bm25		qld	
query set	domain	#q	#documents	avg $\|q\|$	$\|\mathcal{J}.\|$	map	mrr	map	mrr
dbpedia [26]	wikipedia	467	4,635,922	5.37	49,280	0.232	0.565	0.292	0.663
robust04 [63]	news	250	528,155	2.76	311,410	0.199	0.667	0.201	0.681
antique [25]	non-factoid questions	200	403,666	9.34	6,589	0.353	0.881	0.252	0.729
gov2 [10]	*.gov web	150	1,247,753	3.13	135,352	0.157	0.718	0.165	0.706
clueweb09b [9]	web	200	50,000,000	2.45	84,366	0.078	0.383	0.073	0.304

5 Experiment

In this section, we explore the following research questions:

RQ1: How does rrf-fusion perform across different query reformulation?

RQ2: Is the effectiveness of rrf-fusion consistent across diverse datasets?

RQ3: What is the impact of the parameter k on rrf-fusion?

5.1 Dataset

We used well-known query sets in english from different domains, namely dbpedia [26] collection of wikipedia articles, robust04 [63] collection of news articles and US government publications, antique's test collection [25] including open-domain non-factoid questions from Yahoo! Answers, gov2 [10] webpages of .gov web domain, and clueweb09b [9] collection of webpages. In all query sets, we filter out queries with *no* relevance judgment. Also, given an information retrieval method and an evaluation metric, we exclude those original queries that result in the best metric value of 1.00, for no reformulation is needed. Table 1 summarizes the statistics of the query sets. As seen in the robust04, gov2, and clueweb09b query sets, the average query lengths are 2.76, 3.13, and 2.45, respectively, indicating relatively short queries. Conversely, the antique query set exhibits longer queries, with an average length of 9.34 terms, suggesting more detailed or complex information needs. The dbpedia query set falls within an intermediate range, with an average of 5.37 terms.

5.2 Baseline

We compared query backtranslation with 22 existing unsupervised query reformulation methods [60] categorized into two groups, local and global, on five datasets. Global methods consider an original query only and include:

- tagme [18], which replaces the original query's terms with the title of their wikipedia articles,

- stemmers, which utilize various lexical, syntactic, and semantic aspects of query terms and their relationships to reduce the terms to their roots, including `krovetz`, `lovins`, `paiceHusk`, `porter`, `sremoval`, `trunc4`, and `trunc5` [55],
- semantic refiners, which use an external linguistic knowledge-base including `thesaurus` [57], `wordnet` [44], and `conceptnet` [1], to extract related terms to the original query's terms,
- `sense-disambiguation` [61], which resolves the ambiguity of polysemous terms in the original query based on the surrounding terms and then adds the synonyms of the query terms as the related terms,
- embedding-based methods, which use pre-trained term embeddings from `glove` and `word2vec` [39] to find the most similar terms to the query terms,
- `anchor` [30], which is similar to embedding methods where the embeddings trained on wikipedia articles' *anchors*, presuming an anchor is a concise summary of the content in the linked page,
- `wiki` [3], which uses the embeddings trained on wikipedia's hierarchical categories [35] to add the most similar concepts to each query term.
- `backtranslation`, wherein a query is translated from its original language (e.g., `english`) to a set of target languages (e.g., `farsi`, `chinese`, ...) from different language families and cultures, including low-resource languages, and then translate it back to the original language.

Local methods, however, consider terms from top-k retrieved documents via a prior retrieval using an information retrieval method, e.g., `bm25` or `qld`, to find an initial set of most relevant documents among which similar/related terms would be added to an original query. This category includes:

- `relevance-feedback` [54], wherein important terms from the top-k retrieved documents are added to the original query based on metrics like `tf-idf`,
- clustering techniques including `termluster` [7], `docluster` [32], and `conceptluster` [43], where a graph clustering method like Louvain [6] are employed on a graph whose nodes are the terms and edges are the terms' pairwise co-occurrence counts so that each cluster would comprise frequently co-occurring terms. Subsequently, to refine the original query, the related terms are chosen from the clusters to which the initial query terms belong.
- `bertqe` [68], which employs `bert`'s contextualized word embeddings of terms in the top-k retrieved documents.

5.3 Setup

The implementation details of our approach in each of its phases are as follows.

Query Backtranslation. We leverage Meta's *'no language left behind'* (`nllb`) [62], for being open-source, capable of providing two-way translations in 200 languages with a focus on low-resource languages, and easily integrated

Table 2. Languages and their families, alongside the translation quality comparison between nllb and bing. Backtranslation into english is tested to ensure optimal translation quality in the pipeline.

family	language	dbpedia				robust04				antique				gov2				clueweb09b			
		declutr [21]		rouge-l		declutr [21]		rouge-l		declutr [21]		rouge-l		declutr [21]		rouge-l		declutr [21]		rouge-l	
		nllb	bing	nllb	bing	nllb	bing	nllb	bing	nllb	bing	nllb	bing	nllb	bing	nllb	bing	nllb	bing	nllb	bing
	english	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
indo-european	farsi	0.83	0.85	0.62	0.75	0.81	0.85	0.52	0.72	0.84	0.86	0.63	0.76	0.79	0.86	0.47	0.70	0.74	0.80	0.54	0.73
	french	0.87	0.86	0.70	0.81	0.85	0.86	0.56	0.75	0.89	0.89	0.72	0.81	0.82	0.87	0.52	0.75	0.81	0.83	0.60	0.84
	german	0.85	0.87	0.72	0.83	0.81	0.86	0.54	0.74	0.89	0.89	0.73	0.82	0.79	0.87	0.53	0.73	0.75	0.83	0.59	0.83
	russian	0.86	0.86	0.69	0.79	0.84	0.85	0.56	0.70	0.88	0.86	0.69	0.78	0.81	0.86	0.49	0.68	0.77	0.82	0.54	0.79
austronesian	malay	0.88	0.88	0.69	0.77	0.85	0.88	0.57	0.70	0.88	0.90	0.70	0.81	0.85	0.90	0.53	0.70	0.82	0.84	0.63	0.80
dravidian	tamil	0.84	0.86	0.62	0.81	0.81	0.87	0.50	0.75	0.86	0.87	0.64	0.76	0.82	0.88	0.49	0.79	0.77	0.82	0.56	0.85
bantu	swahili	0.87	0.87	0.69	0.77	0.82	0.86	0.49	0.67	0.88	0.87	0.68	0.76	0.79	0.90	0.44	0.76	0.81	0.84	0.59	0.80
sino-tibetan	chinese	0.80	0.86	0.51	0.71	0.78	0.87	0.45	0.69	0.84	0.86	0.59	0.73	0.77	0.87	0.43	0.64	0.72	0.82	0.42	0.70
koreanic	korean	0.82	0.85	0.58	0.73	0.80	0.84	0.47	0.70	0.83	0.87	0.59	0.75	0.78	0.86	0.43	0.68	0.74	0.81	0.53	0.74
afro-asiatic	arabic	0.83	0.87	0.65	0.77	0.78	0.86	0.53	0.74	0.86	0.87	0.68	0.79	0.77	0.87	0.46	0.69	0.72	0.83	0.51	0.82

into any pipeline with few lines of code. Meta's nllb is available with model card and is developed based on a conditional mixture of several transformers that is trained on data tailored for low-resource languages. On the other extreme, we alternatively chose the bing translator[2], a cloud-based *closed*-source machine translation service offered by Microsoft [38] which supports around 128 languages, yet has *no* publicly available model card and/or documentation, to the best of our search. We deliberately aim to compare the efficacy of our method via two extremes of a well-documented translator against a relatively opaque/obscure translator. We translate queries from english into 10 languages from 7 language families, including malay, swahili, and tamil as low-resource languages.

Table 2 shows the average pairwise similarities between a query and its backtranslated versions using rouge-l and declutr [21]. Backtranslation from english to itself has been performed for unit test purposes where all the results for declutr and rouge-l are expected to be the highest possible value of 1.0 with a negligible change in query length. As seen, all languages could expand the original queries of query sets with new terms in the backtranslated versions with an exception in antique query set where queries are long questions and backtranslation versions are of the same or contracted lengths, while the semantics remained almost surely intact in terms of rouge-l and declutr scores. In terms of translation quality, while rouge-l measures the overlap of n-grams between a pair of an original and backtranslated query, and hence, falls short of capturing topic drifts, if any, declutr relies on the cosine similarity between a pair of query embeddings in a *latent space* and is more effective in measuring semantic similarities. Comparing nllb and bing, while both translators obtain similar performance in terms of the declutr, bing has higher values of rouge-l indicating *fewer* new terms and *less* diverse paraphrases in backtranslated queries, which yield its poorer performance for query refinement task.

[2] https://www.bing.com/Translator.

RAG-Based Retrieval. We integrated rrf [11][3] for the fusion and re-ranking process. We selected this function because it is simpler and more efficient than other fusion metrics, as it merges ranks without depending on arbitrary scores from specific ranking methods. It functions without requiring a special voting algorithm or global information, allowing ranks to be calculated and combined one system at a time, thus eliminating the need to store all rankings in memory. It utilizes the diversity within individual rankings more effectively, allowing a document, ranked highly by a few systems, to significantly improve its overall rank. Moreover, it prevents a simple majority of weak preferences from overshadowing stronger ones, unlike other fusion metrics [11]. For a more accurate comparison, we calculate the rrf metric for groups of documents based on the refiner that generated the query that retrieved the document.

Our approach starts by grouping retrieved documents by docid and qid. We then iterate through these groups, calculating a relevance score for each document based on its rank within the group. This score incorporates a positive constant k for normalization or to regulate the impact of rank on the score. We chose to set k to 60 based on our findings indicating that optimal performance is achieved with a small value.

Search and Evaluation. We have applied two information retrieval methods, namely bm25 [51] and qld [46], using pyserini [36] to retrieve relevant content for the original queries as well as the backtranslated versions and evaluate the retrieval performances based on metrics, including map, mrr, and ndcg, using trec_eval [45]. In total, we create a system to retrieve the most relevant documents for the user. A similar trend is observed for qld. However, due to space constraints, the results for qld can be accessed on our github.

5.4 Results

In response to **RQ1**, we generated query variations using distinct unsupervised methods. We further fused the retrieved documents by these query variations according to our five distinct categories: .all (considering all expanders), .global (only the global expanders), .local (only the local expanders), bt (backtranslations using nllb and bing as expanders), and bt.nllb (considering backtranslations only from nllb translator). This structure ensures that we can thoroughly evaluate the performance and efficacy of each refinement method. We evaluate the results of rrf and non-fused using map, ndcg, and mrr. Each evaluation was compared against the original query evaluation to identify enhancements. In instances where multiple reformulated queries improved the original query, we only considered the best result among them. Table 3 represents the results for all five datasets for bm25.map. Overall, the rrf-based methods exhibit strong performance, with the rrf.all category often achieves the highest improvement percentages. This suggests, as expected, that incorporating a diverse set of reformulated queries tends to yield substantial performance gains.

[3] https://github.com/Raudaschl/rag-fusion.

Table 3. rrf-fused vs. non-fused results.

		bm25.map									
		dbpedia		robust04		antique		gov2		clueweb09	
	reformulation method	$\#q^{**}$	%	$\#q^{**}$	%	$\#q^{**}$	%	$\#q^{**}$	%	$\#q^{**}$	%
rrf	rrf.all	**52**	11.13	33	13.25	17	8.50	**56**	37.58	**41**	20.81
	rrf.global	44	9.42	18	7.23	18	9.00	7	4.70	25	12.69
	rrf.local	37	7.92	12	4.82	38	19.00	18	12.08	8	4.06
	rrf.bt	21	4.50	9	3.61	0	0.00	8	5.37	6	3.05
	rrf.bt.nllb	12	2.57	11	4.42	0	0.00	1	0.67	6	3.05
global	tagmee	49	10.49	9	3.61	11	5.50	5	3.36	10	5.08
	bt.nllb	40	8.57	27	10.84	8	4.00	7	4.70	9	4.57
	wiki	23	4.93	12	4.82	0	0.00	5	3.36	8	4.06
	thesaurus	22	4.71	0	0.00	**72**	36.00	0	0.00	0	0.00
	bt.bing	19	4.07	11	4.42	5	2.50	4	2.68	4	2.03
	sensedisambiguation	17	3.64	10	4.02	3	1.50	0	0.00	10	5.08
	word2vec	17	3.64	7	2.81	3	1.50	1	0.67	3	1.52
	wordnet	12	2.57	5	2.01	1	0.50	1	0.67	3	1.52
	conceptnet	9	1.93	9	3.61	1	0.50	4	2.68	5	2.54
	glove	8	1.71	7	2.81	0	0.00	6	4.03	3	1.52
	stem.lovins	3	0.64	3	1.20	0	0.00	0	0.00	0	0.00
	anchor	2	0.43	2	0.80	2	1.00	2	1.34	2	1.02
	stem.porter	2	0.43	1	0.40	4	2.00	0	0.00	0	0.00
	stem.trunc5	2	0.43	3	1.20	0	0.00	2	1.34	1	0.51
	stem.paicehusk	2	0.43	1	0.40	0	0.00	1	0.67	0	0.00
	stem.trunc4	1	0.21	1	0.40	0	0.00	0	0.00	0	0.00
	stem.krovetz	0	0.00	0	0.00	1	0.50	1	0.67	0	0.00
local	relevance-feedback	16	3.43	**35**	14.06	3	1.50	3	2.01	12	6.09
	rm3	11	2.36	1	0.40	6	3.00	7	4.70	2	1.02
	bertqe	4	0.86	2	0.80	0	0.00	1	0.67	2	1.02
	conceptluster	4	0.86	1	0.40	0	0.00	1	0.67	6	3.05
	docluster	0	0.00	0	0.00	0	0.00	2	1.34	1	0.51
	termluster	0	0.00	0	0.00	0	0.00	5	3.36	2	1.02
original query q		15	3.21	7	2.81	2	1.00	1	0.67	25	12.69
sum		467	100	249	100	200	100	149	100	198	100

It also indicates that combining all reformulated queries enhances the retrieval effectiveness. While rrf.global and rrf.local also show competitive performance, they are generally outperformed by rrf.all, highlighting the advantage of using a holistic set of reformulated queries. Dataset-specific observations further emphasize the benefits of the rrf.all approach. For instance, in the gov2 dataset, rrf.all achieves the highest improvement in both metrics, while for the antique dataset, rrf.local achieves a higher percentage increase in the mentioned metrics, underscoring the effectiveness of local reformulation methods in certain contexts. As previously mentioned, the antique dataset comprises open-domain non-factoid questions, characterized by lengthy queries where each question addresses a specific issue. Local methods enhance these queries by considering terms from the top retrieved documents from an initial retrieval, refining them according to their specific topic. Combining these local methods yields better results than the all category. Translation-based reformulated queries, represented by rrf.bt and rrf.bt.nllb, show less improvement compared to the combined approach, suggesting that while translation-based reformulated queries contribute positively, their impact may be limited when used in isolation. Therefore, integrating them with other reformulated queries can potentially enhance their effectiveness.

To address **RQ2**, we evaluated the fused results from previous experiments and compared them to documents retrieved by the original query across five datasets. Table 4 shows the results of comparing our categories with the original. The datasets span different domains, including news articles and non-factoid questions. Across all datasets, the rrf-based methods generally outperformed the original query results. The methods showed a clear trend of higher efficacy, particularly noticeable with the rrf.all and rrf.local. These categories frequently achieved the highest or second-highest scores across various metrics, indicating an improvement in retrieval performance. When analyzing the performance across different query lengths, the rrf-based methods demonstrated more success with longer queries. In datasets with longer average query lengths, such as antique, which has an average query length of 9.34 terms, the improvement was particularly significant. The complexity and detail in longer queries benefited more from the diverse retrieval approaches of the rrf-based methods. In contrast, for datasets with shorter average query lengths, such as robust04 (average query length of 2.76 terms) and clueweb09b (average query length of 2.45 terms), the improvement was present but less pronounced. The shorter queries, which are often more straightforward, did not leverage the full potential of the rrf-based methods as effectively as longer, more complex queries did. Comparing the retrieved documents from these methods to those from the original query, the rrf-based methods consistently retrieved more relevant documents and achieved higher average scores. This improvement suggests that the rrf approach provides a more detailed and comprehensive retrieval process, capturing a broader range of relevant information. Among the different categories, rrf.all emerged as the most successful. This category, which considers all documents retrieved for all query variations, consistently achieved the highest scores across various metrics. The broad and inclusive nature of this method likely contributed to its success, as it combines the strengths of multiple query expansions and retrieval strategies, leading to a more effective overall retrieval process.

To answer **RQ3** and observe the effect of the constant k in the rrf, we conducted multiple experiments across different values in {0, 10, 20, ... 100}.

Table 4. Comparison of the efficacy of `rrf`-fused and original query.

		dbpedia			robust04			antique			gov2			clueweb09		
		#q∗	%	**avg**	#q∗	%	**avg**	#q∗	%	**avg**	#q∗	%	**avg**	#q∗	%	**avg**
	original	23	4.93	0.232	14	5.62	0.199	9	4.50	0.353	1	0.67	0.157	29	14.65	0.078
bm25.map	rrf.all	**96**	20.56	0.289	**62**	24.90	0.223	37	18.50	0.404	**71**	47.65	0.231	**62**	31.31	0.088
	rrf.global	88	18.84	0.241	38	15.26	0.211	24	12.00	0.350	14	9.40	0.167	39	19.70	0.057
	rrf.local	87	18.63	0.210	46	18.47	0.183	**107**	53.50	0.239	36	24.16	0.131	21	10.61	0.051
	rrf.bt	48	10.28	0.258	22	8.84	0.220	1	0.50	0.446	17	11.41	0.214	13	6.57	0.065
	rrf.bt.nllb	28	6.00	0.234	19	7.63	0.197	1	0.50	0.240	4	2.68	0.164	14	7.07	0.067

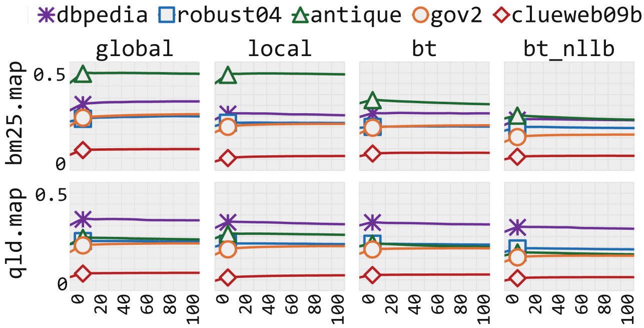

Fig. 2. Effect of constant k on fusion outcomes across various categories.

From Fig. 2 as expected from the results of `rrf`, the experiments indicated that k equal to 60 was near-optimal, though the choice of k was not critically sensitive. This suggests that while k is an important parameter, the robustness of `rrf` in providing high-quality rankings remains consistent across a range of k values, reinforcing its utility in various contexts.

6 Concluding Remarks

In this paper, we proposed backtranslation as an unsupervised method to enhance the retrieval phase of retrieval-augmented generation (rag) systems. We showed that query backtranslation creates diverse and semantically enriched variations of the original query without semantic drift and, hence, could improve the retrieval phase of rag systems. Our experiment demonstrated that (1) fusion methods outperform other query reformulation methods. Specifically, query backtranslation demonstrated substantial performance gains. (2) The efficacy of `rrf` is consistent across diverse datasets. Our future research includes studying the effect of these improved retrieved documents on the generation phase. Further, we will explore the effectiveness of other fusion metrics.

References

1. ConceptNet. http://conceptnet.io/
2. Ahmad, W.U., Chang, K.W., Wang, H.: Context attentive document ranking and query suggestion. In: SIGIR, pp. 385–394 (2019)
3. Al-Shboul, B., Myaeng, S.H.: Wikipedia-based query phrase expansion in patent class search. Inf. Retrieval **17**, 430–451 (2014)
4. Arabzadeh, N., Bigdeli, A., Seyedsalehi, S., Zihayat, M., Bagheri, E.: Matches made in heaven: toolkit and large-scale datasets for supervised query reformulation, pp. 4417–4425 (2021)
5. Bhaisaheb, S., Paliwal, S., Patil, R., Patwardhan, M., Vig, L., Shroff, G.: Program synthesis for complex QA on charts via probabilistic grammar based filtered iterative back-translation. In: EACL 2023, pp. 2501–2515 (2023)
6. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. P10008 (2008)
7. Carpineto, C., De Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. TOIS **19**(1), 1–27 (2001)
8. Chouhan, A., Gertz, M.: Lexdrafter: terminology drafting for legislative documents using retrieval augmented generation. In: LREC/COLING, pp. 10448–10458 (2024)
9. Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the trec 2009 web track. In: TREC (2009)
10. Clarke, C.L., Scholer, F., Soboroff, I.: The trec 2005 terabyte track. In: TREC (2005)
11. Cormack, G.V., Clarke, C.L., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: SIGIR, pp. 758–759 (2009)
12. Dehghani, M., Rothe, S., Alfonseca, E., Fleury, P.: Learning to attend, copy, and generate for session-based query suggestion. In: CIKM, pp. 1747–1756 (2017)
13. Diaz, F.: Condensed list relevance models. In: Proceedings of the 2015 International Conference on The Theory of Information Retrieval, pp. 313–316 (2015)
14. Fabbri, A., et al.: Improving zero and few-shot abstractive summarization with intermediate fine-tuning and data augmentation. In: NAACL, pp. 704–717 (2021)
15. Fan, R., Fan, Y., Chen, J., Guo, J., Zhang, R., Cheng, X.: RIGHT: retrieval-augmented generation for mainstream hashtag recommendation. In: ECIR (2024)
16. Fayyazi, R., Taghdimi, R., Yang, S.J.: Advancing TTP analysis: harnessing the power of encoder-only and decoder-only language models with retrieval augmented generation. CoRR (2024)
17. Feng, R., Hong, X., Jobanputra, M., Warning, M., Demberg, V.: Retrieval-augmented modular prompt tuning for low-resource data-to-text generation. In: LREC/COLING (2024)
18. Ferragina, P., Scaiella, U.: Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In: CIKM, New York, NY, USA, pp. 1625–1628 (2010)
19. Friederici, A.D.: Language in Our Brain: The Origins of a Uniquely Human Capacity. MIT Press (2017)
20. Fu, H., et al.: Doc2bot: accessing heterogeneous documents via conversational bots. In: EMNLP (2022)
21. Giorgi, J.M., Nitski, O., Wang, B., Bader, G.D.: Declutr: deep contrastive learning for unsupervised textual representations. In: ACL/IJCNLP (2021)

22. Glass, M.R., Rossiello, G., Chowdhury, M.F.M., Naik, A., Cai, P., Gliozzo, A.: Re2g: retrieve, rerank, generate. In: NAACL (2022)
23. Hall, J.K.: Teaching and researching: Language and culture. Routledge (2013)
24. ul Haq, S., Abdul-Rauf, S., Shaukat, A., Saeed, A.: Document level NMT of low-resource languages with backtranslation. In: EMNLP (2020)
25. Hashemi, H., Aliannejadi, M., Zamani, H., Croft, W.B.: Antique: a non-factoid question answering benchmark. In: ECIR (2020)
26. Hasibi, F., Nikolaev, F., Xiong, C., Balog, K., Bratsberg, S.E., Kotov, A., Callan, J.: DBpedia-entity v2: a test collection for entity search. In: SIGIR (2017)
27. Hemmatizadeh, F., Wong, C., Yu, A., Fani, H.: Latent aspect detection via back-translation augmentation. In: CIKM (2023)
28. Hu, X., et al.: A systematic view of model leakage risks in deep neural network systems. IEEE Trans. Comput. (2022)
29. Ibrahim, M., Torki, M., El-Makky, N.M.: Alexu-backtranslation-tl at semeval-2020 task 12: improving offensive language detection using data augmentation and transfer learning, pp. 1881–1890. COLING (2020)
30. Kraft, R., Zien, J.: Mining anchor text for query refinement. In: WWW (2004)
31. Kresevic, S., Giuffrè, M., Ajcevic, M., Accardo, A., Crocè, L.S., Shung, D.L.: Optimization of hepatological clinical guidelines interpretation by large language models: a retrieval augmented generation-based framework. NPJ Digit. Medicine (2024)
32. Lee, K.S., Croft, W.B., Allan, J.: A cluster-based resampling method for pseudo-relevance feedback. In: SIGIR (2008)
33. Lewis, P.S.H., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: NeurIPS (2020)
34. Li, Y., Li, X., Yang, Y., Dong, R.: A diverse data augmentation strategy for low-resource neural machine translation. Information **11**(5), 255 (2020)
35. Li, Y., Zheng, R., Tian, T., Hu, Z., Iyer, R., Sycara, K.P.: Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In: COLING (2016)
36. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: a Python toolkit for reproducible information retrieval research with sparse and dense representations. In: SIGIR (2021)
37. Mao, X., Huang, S., Li, R., Shen, L.: Automatic keywords extraction based on co-occurrence and semantic relationships between words. IEEE Access **8**, 117528–117538 (2020)
38. Microsoft: Azure AI custom translator neural dictionary delivering higher terminology translation quality (2023)
39. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, **26** (2013)
40. Moon, J., Cho, H., Park, E.L.: Revisiting round-trip translation for quality estimation. In: EAMT (2020)
41. Nakano, R., et al.: WebGPT: browser-assisted question-answering with human feedback. CoRR (2021)
42. Narayanan, Y.L., Fani, H.: Repair: an extensible toolkit to generate large-scale datasets via transformers for query refinement. In: CIKM (2023)
43. Natsev, A., Haubold, A., Tešić, J., Xie, L., Yan, R.: Semantic concept-based query expansion and re-ranking for multimedia retrieval. In: MM (2007)
44. Pal, D., Mitra, M., Datta, K.: Improving query expansion using wordnet. JASIST **65** (2014)

45. Palotti, J., Scells, H., Zuccon, G.: Trectools: an open-source python library for information retrieval practitioners involved in trec-like campaigns. In: SIGIR (2019)
46. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: SIGIR (2017)
47. Rackauckas, Z.: Rag-fusion: a new take on retrieval-augmented generation. CoRR (2024)
48. Rajaei, D., Taheri, Z., Fani, H.: No query left behind: Query refinement via backtranslation. In: CIKM. Springer (2024)
49. Ramakrishna, A., Gupta, R., Lehmann, J., Ziyadi, M.: INVITE: a testbed of automatically generated invalid questions to evaluate large language models for hallucinations. In: EMNLP, pp. 5422–5429. Association for Computational Linguistics (2023)
50. Raunak, V., Menezes, A., Junczys-Dowmunt, M.: The curious case of hallucinations in neural machine translation. In: NAACL-HLT (2021)
51. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009)
52. Rocchio, J.J.: Relevance feedback in information retrieval. In: The Smart Retrieval System - Experiments in Automatic Document Processing, pp. 313–323 (1971)
53. Sachan, D.S., et al.: End-to-end training of neural retrievers for open-domain question answering. In: ACL/IJCNLP (2021)
54. Salton, G.: The SMART retrieval system–experiments in automatic document processing. Prentice-Hall, Inc. (1971)
55. Schofield, A., Mimno, D.: Comparing apples to apple: the effects of stemmers on topic models. TACL **4** (2016)
56. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green AI. Commun. ACM **63**(12), 54–63 (2020)
57. Shiri, A.A.: End-user interaction with thesaurus-enhanced search interfaces, an evaluation of search term selection for query expansion (2003)
58. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: EMNLP (2021)
59. Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., Nie, J.Y.: A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In: CIKM (2015)
60. Tamannaee, M., Fani, H., Zarrinkalam, F., Samouh, J., Paydar, S., Bagheri, E.: Reque: a configurable workflow and dataset collection for query refinement. In: CIKM (2020)
61. Tan, L.: PYWSD: python implementations of word sense disambiguation (WSD) technologies [software] (2014)
62. Team, N., et al.: No language left behind: scaling human-centered machine translation. línea]. Disponible en: https://github.com/facebookresearch/fairseq/tree/nllb (2022)
63. Voorhees, E.: Overview of the trec 2004 robust retrieval track (2005-08-01 2005)
64. Wang, Y., Wang, W., Joty, S., Hoi, S.C.: CodeT5: identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In: EMNLP (2021)
65. Xu, J., Croft, W.B.: Quary expansion using local and global document analysis. In: ACM SIGIR Forum, New York, NY, USA, vol. 51, pp. 168–175. ACM (2017)
66. Yan, S., Gu, J., Zhu, Y., Ling, Z.: Corrective retrieval augmented generation. CoRR (2024)

67. Zheng, Z., Hui, K., He, B., Han, X., Sun, L., Yates, A.: Bert-QE: contextualized query expansion for document re-ranking. arXiv preprint arXiv:2009.07258 (2020)
68. Zheng, Z., Hui, K., He, B., Han, X., Sun, L., Yates, A.: BERT-QE: contextualized query expansion for document re-ranking. In: EMNLP (2020)
69. Zhuo, T.Y., Xu, Q., He, X., Cohn, T.: Rethinking round-trip translation for machine translation evaluation. In: ACL, pp. 319–337 (2023)