Re3val: Reinforced and Reranked Generative Retrieval

Anonymous ACL submission

Abstract

Generative retrieval models encode pointers to information in a corpus as an index within the model's parameters. These models serve as part of a larger pipeline, where retrieved information conditions generation for knowledge-intensive NLP tasks. However, we identify two limitations: the generative retrieval does not account for contextual information. Secondly, the retrieval can't be tuned for the downstream readers as decoding the page title is a non-differentiable operation. This paper introduces Re3val, trained with generative reranking and reinforcement learning using limited data. Re3val leverages context acquired via Dense Passage Retrieval to rerank the retrieved page titles and utilizes REINFORCE to maximize rewards generated by constrained decoding. Additionally, we generate questions from our pre-training dataset to mitigate epistemic uncertainty and bridge the domain gap between the pre-training and fine-tuning datasets. Subsequently, we extract and rerank contexts from the KILT database using the rerank page titles. Upon grounding the top five reranked contexts, Re3val demonstrates the Top 1 KILT scores compared to all other generative retrieval models across five KILT datasets.

1 Introduction

The primary objective of retrieval models is to enhance the accuracy of answers by selecting the most relevant documents retrieved for a given query, ensuring models have sufficient information to help the downstream reasoning process. For instance, DRQA (Chen et al., 2017) introduces a "retrieve and read" pipeline using TF-IDF to return documents for a question answering model to achieve this goal. More recently, NLP researchers have studied neural retrieval models like Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) with a seq2seq model to build retrieval augmented language models.

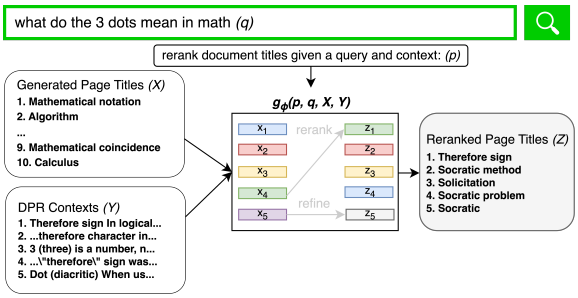


Figure 1: Re3val's Page Title Reranker (g_ϕ) utilizes contexts (Y) from DPR to rerank and refine generated page titles (X), resulting in reranked page titles (Z).

Rather than using inner-product based retrieval, generative retrieval models such as GENRE (Cao et al., 2021) and CorpusBrain (Chen et al., 2022) generate page titles through constrained decoding, attaining higher R-Precision and Recall compared to DPR. In our work, we further evaluate how additional contextual information can benefit the generative retrieval models through reranking and how reinforcement learning can enhance relevance through reward signals.

We introduce Re3val: Reinforced and Reranked Generative Retrieval, a novel framework specifically designed to address the challenges in neural information retrieval. Our approach utilizes 500k pre-training data and 48k task-specific data for training. Despite the reduced data used in distant supervision, Re3val achieves exceptional performance. Our contributions are described as below:

- We combine the initially retrieved page titles with contexts obtained from DPR, facilitating the generative reranking process of the page titles. Through this reranking procedure, Re3val outperforms other generative retrieval models including GENRE, CorpusBrain, and SEAL (Bevilacqua et al., 2022), in terms of average R-Precision across five tasks, showcasing an average increase of 1.9%.

- We incorporate REINFORCE (Williams, 1992) to facilitate information integration during the decoding process of generative retrieval. Combined with question generation, REINFORCE enables Re3val to outperform CorpusBrain zero-shot retrieval with an average improvement of 8% in R-Precision across five tasks.

- We suggest a new retrieval pipeline that extracts the contexts for the reranked page titles, applies our context reranker, and grounds answers with the reranked contexts. As a result, Re3val distinguishes itself by achieving the highest KILT scores among other generative retrieval models, with an average increase of 2.1%.

In summary, Re3val uses DPR contexts for reranking page titles, leading to improved R-Precision. Re3val enhances performance by integrating generated questions in pre-training and utilizing REINFORCE during distant supervision. Moreover, Re3val achieves more accurate answers by reading reranked contexts retrieved with the reranked page titles. These advancements enable Re3val to achieve state-of-the-art performance while also offering cost savings by reducing training time and minimizing the need for extensive data labeling.

2 Related Work

2.1 Document Retrieval

TF-IDF (Johns, 1972) and BM25 (Robertson et al., 2009) assign weight to terms in a document based on their term frequency and inverse document frequency. These methods cannot inherently consider semantic shift or distribution similarity while computing similarity metrics. In light of this limitation, Karpukhin et al. (2020) introduce the Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), establishing a bi-encoder that creates dense embeddings of questions and related passages within a corpus. These embeddings are subsequently compared using a dot product operation. During inference, DPR retrieves the top-k relevant contexts employing either Nearest Neighbor Search or Maximum Inner Product Search on the FAISS index. Guu et al. (2020) and Lewis et al. (2020) retrieve knowledge from a corpus using DPR and generate an answer using a variant of the Transformer models. FiD (Fusion in Decoder) (Izacard and Grave, 2021) extends T5 (Wolf et al., 2020) by combining independently encoded queries and retrieved passages to decode an answer. However, these models do not rerank retrieved documents that allow a reader to perform better with fewer contexts utilized for a reader.

2.2 Generative Retrieval

Cao et al. (2021) introduce an Autoregressive Entity Retrieval model (GENRE). GENRE utilizes seq2seq language models for page title retrieval and employs a trie-based constrained decoding approach. This allows GENRE to assign a probability of 0 to non-existing page titles, ensuring accurate retrieval. Moreover, Chen et al. (2022) propose CorpusBrain, a generative retrieval model encoding the knowledge about the corpus through pre-training strategies. DEARDR (Thorne, 2022) proposes three distinct pre-training regimens and a data-efficient distant supervision method for generative retrieval. Moreover, SEAL (Bevilacqua et al., 2022) leverages an FM-Index to efficiently generate n-grams within the corpus for fast lookup speed without increasing the index size. The Differentiable Search Index (DSI) (Tay et al., 2022) employs a seq2seq model to map individual queries to atomic document identifiers, which in turn are associated with segmented chunks of the document. Similarly, the Neural Corpus Index (NCI) (Wang et al., 2022) utilizes hierarchical k-means for document representation, generates queries based on content, and trains a transformer model with a Prefix-Aware Weight-Adaptive Decoder using Consistency-based regularization. However, these models overlook the opportunity to minimize additional entropies in retrieved page titles or documents by incorporating contextual information. Leveraging such information reduces randomness and refines the ranking. Moreover, these models overlook the potential benefits of harnessing knowledge during decoding.

2.3 Question Generation

In the past, numerous endeavors (Labutov et al., 2015; Chali and Hasan, 2015; Serban et al., 2016; Duan et al., 2017) have been made to generate questions to enhance the task of Question Answering. Recently, studies analyzing questions have attempted to find the relationship with contexts. Mao et al. (2021) propose Generation-Augmented Retrieval (GAR) that generates query contexts. GAR employs a BM-25 retrieval model and achieves per-

formance comparable to DPR. Sachan et al. (2022) create questions based on the retrieved contexts and rerank contexts based on the log-likelihood score over the generated questions. However, these studies overlook the fact that question generation can address the epistemic uncertainty in question answering tasks by minimizing the domain gap between pre-training and fine-tuning data.

2.4 Reranking Models

Reranking in information retrieval involves refining the initial ranking of retrieved documents by utilizing scores from a more complex query, as exemplified by Elastic Search[1]. Atlas (Izacard et al., 2022b) retrieves documents with Contriever (Izacard et al., 2022a), reranks the retrieved documents, and reasons with FiD. Re^{2}G (Glass et al., 2022) employs a cross-encoder (Rosa et al., 2022; Nogueira and Cho, 2020) to rerank retrieved documents based on softmax probability using $BM25(q) \cup DPR(q)$, determining the relevance between a query and context. FiD-Light (Hofstatter et al., 2022) introduces a compression for encoded passages and reranks candidate lists using source pointers. These source pointers are textual indices that represent the relevant context, as initially introduced in FiD-Ex (Lakhotia et al., 2021). However, these reranking models do not perform reranking at the page title level and do not make use of a rerank query.

2.5 Reinforcement Learning

When framing text generation as a Reinforcement Learning (RL) problem, the state (s_t) at time t can be seen as a sequence of tokens. At the same time, the action (a_t) represents the probability distribution of the generated token. This formulation can incorporate non-differentiable feedback, such as common evaluation metrics as reward. Moreover, various RL methodologies such as REINFORCE (Williams, 1992), Advantage Actor-Critic (A2C) (Mnih et al., 2016), and Proximal Policy Optimization (PPO) (Schulman et al., 2017) are being successfully applied in a multitude of scenarios. This study primarily utilizes REINFORCE, a simple yet effective method.

3 Methodology

The primary contribution of Re3val is its capability to generatively rerank page titles by incorporating contextual information and to apply REINFORCE during distant supervision of a generative retrieval. Additionally, Re3val utilizes question generation for pre-training. Furthermore, Re3val pioneers the reading of contexts retrieved using page titles obtained through a generative retrieval approach.

The following step-by-step description of stages elucidates how the methods proposed at each stage, as depicted in Figure 2, are sequentially integrated into the overall pipeline.

3.1 Page Title Retrieval

3.1.1 Pre-training

Following DearDr (Thorne, 2022), we pre-train the generative retriever. To mitigate the domain shift problem during pre-training for question-answering and dialogue tasks, we generate questions for half of the pre-training passages. We utilize Flan-T5 base (Chung et al., 2022) to create questions given a prompt, "Generate a question related to the following Passage: ". Among generated questions, we employ Spacy's Entity Recognizer of en_core_web_sm[2] to filter out ambiguous questions such as "Where is he". Specifically, we remove questions that do not contain entities other than DATE, MONEY, CARDINAL, TIME, QUANTITY, ORDINAL, and PERCENT.

3.1.2 Pre-Training and Few-Shot Training (Stage 1,3)

During the pre-training and fine-tuning of Re3val, an instructive prompt - "rank document titles given a query: " - is introduced before each query on the T5 model (Wolf et al., 2020). In Few-Shot training, we added labeled data to narrow the range of target candidates.

3.1.3 REINFORCE (Stage 2,4)

The REINFORCE is employed during training to optimize the black box of zero-shot (Stage 1) and few-shot (Stage 3) retrieval in Re3val. The REINFORCE utilizes the R-Precision of generated page titles as a reward. The effectiveness of the REINFORCE, along with the formula for gradient computation of the REINFORCE objective function, is demonstrated in Appendix A.5.

3.2 Page Title Reranker (Stage 5-7)

Retrieved page titles are initially ranked based on their relevance score, computed by our retrieval model. Then, a reranking query can be introduced

[1] https://www.elastic.co

[2] https://spacy.io

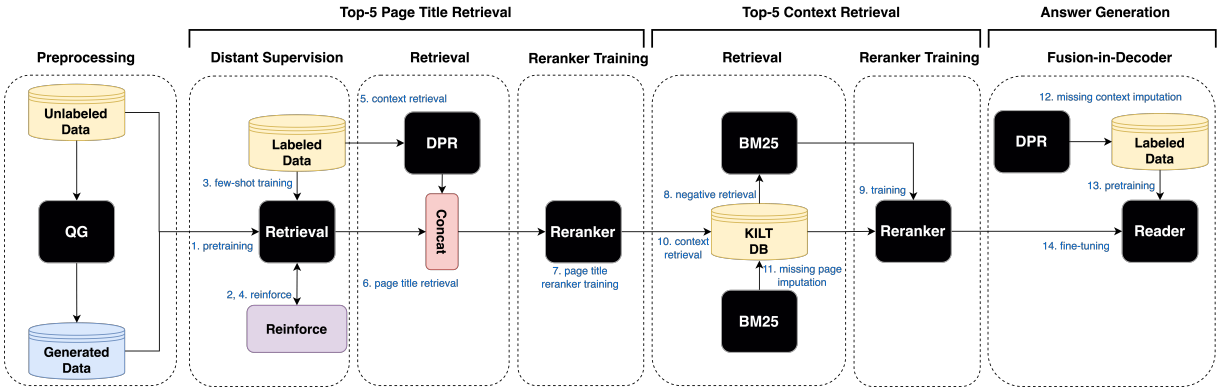


Figure 2: Re3val Training Pipeline. Generated questions after filtering are integrated into pre-training (1), followed by few-shot training (3) with REINFORCE (2, 4). Retrieved DPR contexts (5), perturbed page titles (6), and queries are concatenated for reranker training (7). Gold and negative passages retrieved with BM-25 are employed (8) for context reranker training (9). Contexts are retrieved using the top 5 reranked titles from KILT (10), where missing titles are imputed with BM-25 (11). DPR contexts are imputed (12) if lacking five gold contexts during FiD model pre-training (13). FiD model is fine-tuned using five reranked contexts (14).

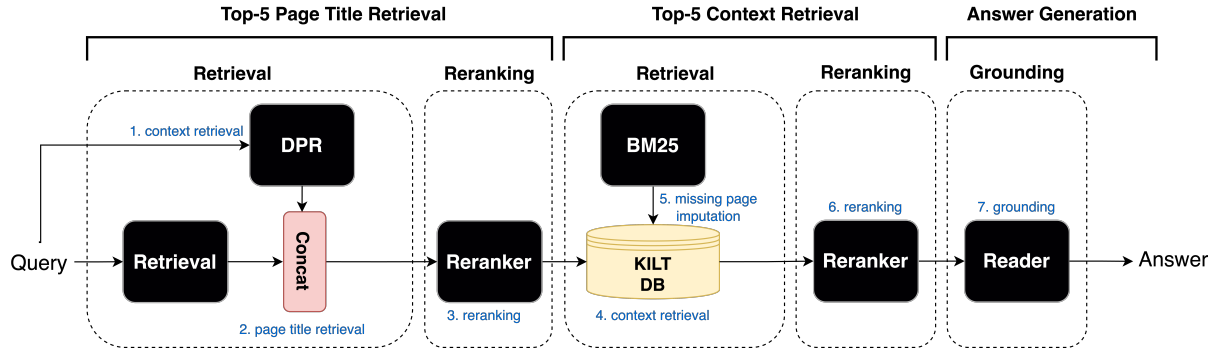


Figure 3: Re3val Inference Pipeline. Reranker concatenates retrieved DPR contexts (1), page titles (2), and query to rerank page titles (3). Contexts retrieved with the top five reranked page titles (4), including BM-25 imputed titles (5), are reranked (6). The top-5 reranked contexts are used to generate an answer (7).

to refine the ranking further and increase the likelihood of obtaining the most relevant page titles. However, the KILT datasets do not provide a specific reranking query.

To address the limitation above, our page title reranker leverages contexts retrieved via an auxiliary index, such as the Dense Passage Retrieval multi-set checkpoint[3], to serve as the reranking query. Unlike the prompt for ranking, which is "rank document titles given a query: ", the prompt for reranking is modified to "rerank document titles given a query and contexts: ".

In order to improve the refinement and reranking functions of our page title reranker, we have implemented a new training strategy. This strategy combines REINFORCED few-shot (Stage 4) and zero-shot (Stage 1) retrieved page titles during training. Additionally, we apply uniform shuffling to the page titles in the top half of the training sets generated by our zero-shot and few-shot retrieval models.

Mixing titles from different checkpoints and shuffling retrieved page titles introduces noise to the input data. This noise is beneficial as it enables the page title reranker to filter out inconsistencies, outliers, and misleading patterns in the test set, ultimately enhancing its performance.

3.3 Context Retrieval (Stage 10-11)

Preprocessing (Stage 10) To refine the data for context retrieval for a reader, we divide each context in the KILT Database into chunks, each consisting of 100 words. To ensure data quality and relevance, we filter out sentences that only contain a page title, as well as sentences containing the specific patterns, "Section::::" or "BULLET::::".

[3] https://github.com/facebookresearch/DPR

Extraction (Stage 10,11) After the page title reranking process, we acquire five reranked page titles. Subsequently, we retrieve the corresponding contexts for each of these page titles. In situations where specific page titles are unavailable in the KILT database, we suggest using the BM-25 imputation method. This method employs the BM-25 algorithm to impute the most suitable page title from the KILT database. A detailed analysis of this imputation approach can be found in Appendix A.6.

3.4 Context Reranker (Stage 8-11)

To enhance the reader's experience, we reduce memory and context usage through our Context Reranker. Specifically, we use a cross-encoder to assess the relevance of a query and context pair for reranking the contexts derived from the five page titles. The input structure for our context reranker is as follows: "[CLS] Query [SEP] Context [SEP]".

We utilize gold passages as positive examples for training and evaluating our Context Reranker. We also include two types of hard negative examples retrieved with BM-25: the top 128 unlabeled context chunks mapped to labeled page titles in the training set and the top 128 unlabeled context chunks mapped to the unlabeled page titles retrieved by our Page Title Reranker.

3.5 Reader (Stage 12-14)

We employ the Fusion in Decoder (FiD) as our reader for the reading task. During the pre-training phase of FiD, we utilize gold passages and impute Dense Passage Retrieval (DPR) contexts for queries with fewer than five available gold contexts. Subsequently, following the pre-training phase, we perform fine-tuning of the FiD model using the top five or ten contexts retrieved by our context reranker.

4 Experiments

4.1 Datasets

We use datasets from the KILT (Petroni et al., 2021) benchmark. We study Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and HotpotQA (Yang et al., 2018) for question answering tasks, FEVER (Thorne et al., 2018) for a fact-checking task, and WoW (Dinan et al., 2018) for a dialogue task, which are publicly available[4]. Comprehensive details about the datasets are discussed in Appendix A.2.

4.2 Evaluation

KILT utilizes a page-level retrieval strategy, and the assessment of page-level retrieval tasks measures the capacity to present a collection of Wikipedia pages as supporting evidence for a prediction, assessed through R-Precision and Recall@k metrics. R-Precision quantifies the proportion of relevant documents retrieved out of the total retrieved documents. On the other hand, Recall@k quantifies the proportion of relevant documents retrieved out of the total number of actual documents, taking into account only the top-k retrieved documents. Downstream reading tasks utilize different evaluation metrics depending on the specific task. For example, question-answering tasks are evaluated using Exact Match (EM) and F1 score. Dialogue tasks employ metrics such as ROUGE-L and F1 score. Fact verification tasks, on the other hand, are evaluated based on Accuracy. However, KILT has recently introduced the KILT score[5] as a ranking metric for evaluating downstream performance. The KILT score takes into account post-processed Accuracy, EM, ROUGE-L, and F1 scores mentioned in Appendix A.8.3, but only if the R-Precision for a given query is 1. For detailed information regarding the metrics for evaluation, please refer to Appendix A.8.

4.3 Retrieval

We conduct our retrieval experiments by initializing three pre-trained models with varying parameters to investigate the impact of model size on performance. Specifically, we employ the following pre-trained models: t5-small, t5-base, and t5-large (Wolf et al., 2020).

Training We utilize 250k uniformly sampled June 2017 and August 2019 Wikipedia dumps for the pre-training phase across all datasets. Additionally, we generate questions from an additional 250k uniformly sampled Wikipedia dumps and include them in the training process. For fine-tuning, we utilize 48k uniformly sampled task-specific datasets. Detailed information about the datasets can be found in Appendix A.2 and Table 8. Importantly, we reinforce the zero and few-shot retrieval stages by employing the same dataset for each retrieval stage.

[4] https://github.com/facebookresearch/KILT

[5] https://eval.ai/web/challenges/challenge-page/689/evaluation

Evaluation We employ a multi-beam search approach with a beam size specified in Table 4 to assess the performance on all development and test sets. In addition, we select the top five page titles from the list of multi-page titles generated per query for evaluation purposes.

4.4 Page Title Reranker

In our experimentation, we explore two types of initialization for our page title reranker. Firstly, we initialize the reranker using the plain t5-small, t5-base, and t5-large models. Secondly, considering the three different model sizes, we utilize the checkpoint from the reinforced few-shot retrieval process. To maintain input compatibility, we limit the query for the reranker's input to the first 250 words. In addition, the input - consisting of a query, ten page titles, and five contexts - is truncated to a maximum of 512 tokens.

4.5 Context Reranker

In our experiments, for all datasets, a query and a context are separated using the special token "[SEP]", and trained using the nboost/pt-bert-base-uncased-msmarco[6] as input. We input the first 150 words of a query for question-answering and fact-verification tasks. In the case of a dialogue task, the last 300 words of the query are used, as the final sentence often serves as the closure to the conversation. The maximum sequence length of input is detailed in Table 4 and 6, providing further information on the specific limitations imposed on the input size.

4.6 Reader

Two types of inputs are used for pre-training our two versions of FiD. The first type includes only gold passages, while the second consists of gold passages and top-ranked Dense Passage Retrieval (DPR) contexts. For the Natural Questions (NQ) dataset, pre-training is conducted using the NQ FiD checkpoint, which has been pre-trained on 770 million parameters[7]. For the remaining datasets, pre-training is performed using the TriviaQA FiD checkpoint, which has been pre-trained on 770 million parameters[7]. Regarding the Wizard of Wikipedia (WoW) dataset, we retain the last 385 words of the query for input. For other datasets, we use the first 125 words. The maximum sequence length is outlined in Table 4 and 6, providing specific details on the constraints imposed on input size.

An example of an input format is "question: query title: page_title, context: retrieved_title". In this format, "question:", "title:", and "context:" are special tokens, while "query", "page_title", and "retrieved_title" represent variables denoting the respective components of the input.

Following the pre-training phase, we conduct fine-tuning, incorporating 5 or 10 contexts retrieved using our Context Reranker.

5 Result

5.1 Retrieval

5.1.1 Zero-shot Retrieval

Based on the findings presented in Table 1, CorpusBrain exhibits an 8% lower R-Precision on average compared to Re3val, despite being trained on more than 500 times more data. We hypothesize that the question-generation process mitigates the epistemic uncertainty resulting from limited training data, thus minimizing the domain shift between the pre-training and task-specific fine-tuning data.

Examining Table 12 in the Appendix, we observe that REINFORCE yields a modest improvement in the performance of zero-shot retrieval, with a few exceptions. Specifically, REINFORCE effectively captures the variability introduced during the constrained beam search exploration, as it utilizes the search results as a reward signal, thereby reducing bias towards the pre-training data in our retrieval model.

5.1.2 Few-shot Retrieval

However, as indicated in Table 12, the effectiveness of REINFORCE diminishes when applied to the few-shot retrieval scenario. In some instances, REINFORCE results in performance degradation across specific datasets. We postulate that this phenomenon can be attributed to the inherent variance associated with Reinforcement Learning. Furthermore, the performance degradation may arise from the exploration-exploitation trade-off during the multi-beam search, where a broad range of solution spaces is explored, potentially leading to a decreased focus on exploitation. For instance, Appendix A.9 shows that the relative performance ranking can be reversed as the number of samples (K) increases.

[6] https://huggingface.co/nboost/pt-bert-base-uncased-msmarco

[7] https://github.com/facebookresearch/FiD

Dataset	Question Answering NQ		TQA		HoPo		Fact Check. FEV		Dial. WoW		Average	
Model	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**
Zero-shot												
TF-IDF	28.10	-	46.40	-	34.10	-	50.90	-	49.00	-	41.70	-
CorpusBrain	28.25	-	42.76	-	**44.84**	-	70.38	-	29.64	-	43.17	-
Re3val$_S$	25.20	29.62	**47.47**	27.53	42.91	23.36	74.99	84.19	52.31	64.28	48.58	45.80
Re3val$_B$	33.24	37.90	47.25	52.88	43.82	**24.79**	76.22	83.42	**56.45**	70.05	51.40	53.81
Re3val$_L$	**34.70**	**41.47**	46.38	**53.01**	43.55	22.77	**78.60**	**85.36**	55.67	**72.77**	**51.78**	**55.07**
Few-shot (48k)												
Re3val$_S$	47.44	49.20	61.28	64.32	47.47	27.53	79.74	84.29	56.90	71.86	58.57	59.44
Re3val$_B$	54.15	55.34	63.80	69.83	50.01	31.47	78.67	82.47	62.00	77.50	61.73	63.32
Re3val$_L$	54.92	55.76	63.89	71.35	49.99	32.81	77.15	79.88	62.84	79.91	61.76	63.94
Full Fine-tuning												
DPR + BART	54.29	65.52	44.49	56.99	25.04	10.40	55.33	74.29	25.48	55.10	40.93	52.46
RAG	59.49	67.06	48.68	57.13	30.59	12.59	61.94	75.55	57.78	74.63	51.70	57.39
GENRE	60.25	61.36	69.16	75.07	51.27	34.03	83.64	88.15	62.88	77.74	65.44	67.27
KGI	63.71	**70.17**	60.49	63.54	-	-	75.60	84.95	55.37	78.45	-	-
SEAL	63.16	68.19	68.36	**76.36**	58.83	**51.03**	81.45	89.56	57.55	78.96	65.87	**72.82**
TABi	62.60	64.95	**70.36**	69.16	53.12	35.48	**84.45**	88.62	59.11	69.10	65.93	65.46
CorpusBrain	60.32	61.21	70.19	75.64	51.80	34.57	84.07	**90.50**	**64.79**	**81.85**	66.23	68.75
Reranking (48k)												
Re3val$_S$	59.63	60.78	59.84	64.43	54.93	38.50	81.22	85.90	56.90*	71.86*	62.50	64.29
Re3val$_B$	64.75	63.05	66.31	71.95	56.65	41.14	81.58	83.27	62.00*	77.50*	66.26	67.38
Re3val$_L$	**66.48**	65.40	68.57	74.48	**59.60**	44.21	82.78	85.71	63.32	79.88	**68.15**	69.94

Table 1: The performance results of the generative and bi-encoder retrieval models on the KILT test sets are presented in the table above. The models that achieve the highest performance are indicated in **bold**, while the second-best models are underlined. In the case of Re3val, it utilizes a reinforced version for Zero-shot and Few-shot (48k) results, while an unreinforced version is used for Reranking (48k) results. The Reranking (48k) involves the page title reranker trained using the Vanilla T5 pre-trained model. The subscript notations used in the table denote the model size, where *S* corresponds to t5-small, *B* represents t5-base, and *L* indicates t5-large. For the WoW dataset, the reported scores reflect the few-shot results, except for Re3val$_L$, representing the best overall result.

5.1.3 Page Title Reranker

The validity of our reranker's input concatenation is supported by the principles of Mutual Information theory (Shannon, 1948). Let's define X as the set of page titles and Y as the set of DPR contexts, where X takes values from $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ and Y takes values from $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$. We denote the probability distribution of X as $P(x)$.

The mutual information between X and Y is denoted as $I(X;Y)$, and it quantifies the amount of shared information between the two variables. It is calculated using the formula:

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \quad (1)$$

By considering the joint probability of DPR contexts and page titles, $I(X;Y)$ allows us to gain insights into the dependency between these two variables. Therefore, our page title reranker leverages this shared information to reduce uncertainty in the ranking of page titles, thus improving the reranking and refinement process.

The results obtained from the dev sets are documented in Table 12. Table 12 indicates that the page title reranker, fine-tuned from the reinforced few-shot retrieval, outperforms the reranker initialized from the T5 pre-trained model when the number of parameters is small. However, the opposite trend is observed as the number of parameters increases. While the knowledge about ranking compensates for the limited capacity to learn complex reranking patterns when the number of parameters is small, prior knowledge about ranking interferes with the reranking function as the number of parameters grows. In essence, ranking and reranking serve distinct purposes. Ranking focuses on sorting relevant documents, while reranking involves permuting the initially ranked documents.

The dialogue task requires more detailed reasoning over textual information than question-answering and fact-verification tasks. Reranking

Dataset	|P|	Question Answering NQ		TQA		HoPo		Fact Check. FEV	Dial. WoW	
Model		K.-EM	K.-F1	K.-EM	K.-F1	K.-EM	K.-F1	K.-AC	K.-RL	K.-F1
Pre-training (48k)										
Re3val	5	36.84	42.27	48.34	51.74	23.25	27.55	70.62	9.74	10.81
Re3val$_I$	5	39.88	45.43	51.08	53.93	23.85	28.11	**73.09**	9.88	11.08
Full Fine-tuning										
SEAL	100	38.78	44.40	50.56	**54.99**	18.06	21.42	71.28	10.45	11.63
RAG	5	32.69	37.91	38.13	40.15	3.21	4.10	53.45	7.59	8.75
KGI	5	36.36	41.83	42.85	46.08	-	-	64.41	10.36	11.79
DPR + BART	5	29.09	42.36	46.19	1.96	2.53	63.94	34.70	5.91	6.96
Few-shot (48k)										
Re3val	5	38.92	45.06	50.05	53.14	23.94	28.26	71.06	11.70	13.46
Re3val	10	40.17	**46.53**	**51.31**	54.46	24.13	28.44	71.08	11.79	13.41
Re3val$_I$	5	**40.44**	46.23	50.41	53.44	**24.33**	28.64	72.78	**12.01**	13.55
Re3val$_I$	10	39.54	45.92	51.00	53.93	24.22	**28.71**	73.02	11.94	**13.57**

Table 2: The final KILT scores of the test sets are reported above, as presented on the KILT Leaderboard. The best-performing models are indicated in **bold**, while the second-best models are underlined. Additionally, the notation *I* denotes the *Imputation* of DPR contexts for missing gold contexts.

with a few parameters does not yield improvements in performance for the WoW test set, as indicated in Table 1. Furthermore, the inconsistency between the test set results in Table 1 and the dev set results in Table 12 for the reranking stage of the 770m, 770m parameter configuration highlights the need for further investigation.

5.2 Context Reranker

The performance of our Context Reranker, evaluated using gold passages and hard negative passages as described in Section 4.5, is presented in Table 3. Notably, our Context Reranker exhibits a higher precision compared to recall. This characteristic shows that the Context Reranker effectively filters out irrelevant and low-quality results, prioritizing accuracy in retrieving relevant documents, even if they may miss some. The high precision score indicates that relevant documents are ranked at the top. However, further investigation is required to examine the trade-off between precision and recall in the Context Reranker for downstream reading tasks.

5.3 Reader

The slight performance difference observed between the reader with 5 and 10 contexts in Table 2 suggests that our context reranker excels in retrieving highly relevant documents at the top, showcasing its exceptional precision. Moreover, our context imputation pre-training strategy is effective, enabling Re3val to outperform SEAL, although SEAL utilizes 100 contexts for grounding with FiD. Finally, as indicated in Table 2, Re3val achieves superior results with only five passages, underscoring the advantages of our approach.

6 Conclusion

This paper presents Re3val, a novel reranking architecture for generative retrieval. Re3val achieves state-of-art performance with question generation, REINFORCE, and reranking. Succinctly, Re3val incorporates question generation to address epistemic uncertainty and domain shift. It utilizes REINFORCE on constrained beam search outputs to enhance exploration. Experimental results demonstrate Re3val's superiority over the CorpusBrain zero-shot baseline, with an average 8% R-Precision improvement across five tasks using reduced pre-training data. Re3val also achieves an average 1.9% R-Precision increase compared to other generative models via page title reranking with limited task-specific data. Moreover, by employing a context reranker before grounding, Re3val achieves top-1 KILT scores among generative retrieval models, showing an average 2.1% improvement across five datasets. Re3val's data-efficient approaches reduce training time and labeling costs, representing notable advancements in generative retrieval.

Limitations

Given this project's time and resource limitations, a comprehensive comparison of REINFORCE with

other Reinforcement Learning algorithms, such as PPO and TRPO, which require more memories for their reference model, was not feasible. Furthermore, the observed disparity between the performance on the development and test sets for both the retrieval and reader components necessitates further investigation. Lastly, it is worth noting that specific labeled page titles in the FEVER dataset are not present in the KILT database, introducing a discrepancy that should be considered.

Ethics Statement

In this study, we utilize datasets obtained from various sources, including Natural Questions, TriviaQA, HotpotQA, FEVER, and Wizard of Wikipedia. These datasets serve as integral components of the KILT benchmark and are derived from the KILT knowledge source, which is based on the August 1st, 2019, Wikipedia dump. In addition to the 2019 Wikipedia dump, we incorporate the June 2017 Wikipedia dump into our pre-training. It is crucial to acknowledge that these datasets may contain instances of incorrect or misconstrued information, which could potentially result in the generation of biased, toxic, or fabricated content. Moreover, the utilization of language models, such as T5, during the training and preprocessing stages introduces the possibility of ethical risks that may be embedded within the internal parameters of these models. Consequently, it is imperative for researchers to exercise caution when employing our paper and the associated outputs and to establish suitable policies to mitigate any potential ethical risks that may arise from the use of these models in real-world production settings.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. In *Advances in Neural Information Processing Systems*, volume 35, pages 31668–31683. Curran Associates, Inc.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Yllias Chali and Sadid A. Hasan. 2015. Towards topic-to-question generation. *Computational Linguistics*, 41(1):1–20.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2022. CorpusBrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and et al. 2022. Scaling instruction-finetuned language models.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *CoRR*, abs/1811.01241.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Sebastian Hofstatter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2022. Fid-light: Efficient and effective retrieval-augmented text generation. *https://arxiv.org/pdf/2209.14290.pdf*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Atlas: Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv*, 2208.

Karen Johns. 1972. A statistical interpretation of term specificity and its application in retrieval.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China. Association for Computational Linguistics.

Kushal Lakhotia, Bhargavi Paranjape, Asish Ghoshal, Scott Yih, Yashar Mehdad, and Srini Iyer. 2021. FiD-ex: Improving sequence-to-sequence models for extractive rationale generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3712–3727, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Volodymyr Mnih, Adria Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. In defense of cross-encoders for zero-shot retrieval.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany. Association for Computational Linguistics.

C. E. Shannon. 1948. A mathematical theory of communication. In *The Bell System Technical Journal*.

Yi Tay, Dehghani Mostafa Tran, Vinh Q., Jianmo Ni, Dara Bahri, and Harsh Mehta. 2022. Transformer memory as a differentiable search index. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, USA.

James Thorne. 2022. Data-efficient auto-regressive document retrieval for fact verification. In *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 44–51, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao1, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia1, Chengmin Chi, Guoshuai Zhao, Zheng Liue, Xing Xie, Hao Allen Sun, Weiwei Deng, Qi Zhang, and Mao Yang. 2022. A neural corpus indexer for document retrieval. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, LA, USA.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameters

The default hyperparameter settings and hardware configurations employed for the overall tasks are outlined in Table 4, with further details provided in Tables 5 to 7. Given the limited hardware resources available in our academic environment, we utilize different GPUs for our models, as specified in Table 5. FiD, which uses ten passages, is trained with half of the batch size indicated in Table 4 and 6.

A.2 Data

The number of data points used for pre-training and fine-tuning the retrieval models for each task are outlined in Table 8. GENRE and CorpusBrain utilize 21 billion data points from the 2019 Wikipedia dump and 9 billion from the Blink dataset. In the case of Re3val pre-training, we use a combination of the June 2017 and August 2019 Wikipedia dumps.

For tasks such as Natural Questions (NQ), Wizard of Wikipedia (WoW), TriviaQA, and FEVER, we pre-train the models using 125,000 samples from the 2017 Wikipedia dump and 125,000 relevant samples from the Wikipedia dump obtained through the Dense Passage Retrieval multi-set checkpoint. An additional 250,000 generated questions from the remaining samples are also included in NQ, WoW, and TriviaQA. For HotpotQA, we use 125,000 original contexts and 125,000 data points from the two Wikipedia dumps, generating questions with the remaining 125,000 original contexts and 125,000 data points from the Wikipedia dumps. All subsets are uniformly sampled.

For the Page Title reranking task, we utilize Hotpot contexts instead of Dense Passage Retrieval (DPR) contexts specifically for HotpotQA. For other tasks, we used the Dense Passage Retrieval multi-set checkpoint.

A.3 Prefix Tree

To construct and search the Prefix Tree for all tasks, we utilize the KILT knowledge source[8]. This knowledge source is employed as the basis for building and performing Trie Node search.

A.4 Constrained Decoding

In contrast to GENRE's constrained decoding(Cao et al., 2021), which predicts a single entity per beam, Re3val decodes a list of page titles per beam similar to DEARDR(Thorne, 2022), as depicted in Figure 4. This approach enables us to capture the variability of related entities, as page titles are mapped to an answer in KILT datasets.

A.5 REINFORCE

This section presents a formal mathematical proof showcasing the optimization achieved by utilizing the REINFORCE algorithm in our retrieval system.

[8] http://dl.fbaipublicfiles.com/KILT/kilt_knowledgesource.json

A.5.1 Notation

Let $J(\theta)$ denote the objective function. In the context of Re3val, T represents the number of retrieved page titles in a beam. The function $R(\tau)$ represents the return, which is the cumulative reward associated with a trajectory τ, defined as a sequence of actions (a) and states (s). Finally, we denote the policy as π with parameter θ, and ∇ represents the gradient operator.

A.5.2 Proof

The formula for computing the gradient of the REINFORCE objective function is given by:

$$\nabla J(\theta) = E_{\pi_\theta}\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t, s_t) R(\tau)\right) \quad (2)$$

Here, t represents the timestep. The objective function (2) guides the policy π_θ towards the direction of the gradient. In equation (2), $R(\tau)$ is a scalar derived from the undifferentiable portion of Re3val, specifically the R-precision calculated using a constrained decoding prefix tree.

Re3val generates a sequence of page titles, represented as τ, based on the policy π. The distribution of action a given a state s is denoted as $\pi_\theta(a|s)$. In the case of Re3val, a softmax function is applied to the cross entropy loss to obtain a probability distribution for the action a. Therefore, the policy parameter can be expressed as:

$$\log \pi_\theta(a_t, s_t) = \sum_{i=1}^{M} y_i \log \bar{y_i} \quad (3)$$

Here, M represents the vocabulary size, which corresponds to the number of unique elements in the vocabulary.

In scenarios where $R(\tau_1) < R(\tau_2)$, the model parameter undergoes a greater number of gradient updates in the direction of $\nabla_\theta(\sum_{j=1}^{M} \log \pi_\theta(a_t, s_t) R(\tau_2))$ compared to $\nabla_\theta(\sum_{j=1}^{M} \log \pi_\theta(a_t, s_t) R(\tau_1))$, provided that $R(\tau_1) > 0$ and $R(\tau_2) > 0$.

Consequently, the REINFORCE enhances the performance of zero-shot and few-shot retrieval by assigning more updates to samples that yield higher rewards, thereby promoting the learning of more relevant patterns and improving overall performance.

A.6 Imputation

A.6.1 Missing Page Imputation

It has been observed that specific page titles retrieved by our model are absent in the KILT database, despite applying the same preprocessing and tokenization procedures to these page titles as those utilized for building the Trie Node. This discrepancy in retrieval is systematically attributed to the labeler's mistake. Notably, as the missingness of top-ranked retrieved page titles can significantly impact performance, we assert that these page titles exhibit Missing Not At Random (MNAR) characteristics.

Let a dataset be $D = \{(x_t^{(i)}, o_t^{(i)})_{t=1}^{T_i}, y^{(i)}\}_{i=1}^{n}$ where x be a page title, o be a missing indicator, y be a relevant context, n be the number of data, T be the number of page titles per a query, f_θ be Re3val's context reranker that produces a logit, and k be the KILT database. For classification, $p(y|x_{1:T}, o_{1:T}, \theta) = \frac{e^{f_\theta(k(x_{1:T}, o_{1:T}))_1}}{\sum_{j=0}^{1} e^{f_\theta(k(x_{1:T}, o_{1:T}))_j}}$. Then, $p(x, o|\theta) = p(x|\theta)p(o|x, \phi)$, indicating missing ($o$) depends on both existing (x) and non-existing (ϕ) page titles in the KILT database. That is, the probability of a missing retrieved page title in the database is related to the page title.

To address this MNAR missingness, we employ the BM-25 algorithm to impute the best matching page title from the KILT database. The outcomes of this imputation strategy are presented in Table 9, illustrating that the performance of our reranker on the test sets improves through the imputation.

A.6.2 Missing Context Imputation

Within the KILT dataset, contexts may be pertinent to an answer but have remained unlabeled due to biases from the labeler. This particular phenomenon aligns with the characteristics of Missing Not At Random (MNAR) since the absence of these contexts is systematically linked to the actions of the labeler. Table 2 demonstrates a notable performance improvement when utilizing imputation techniques to address sparse contexts in a query using the DPR (Dense Passage Retrieval) method.

A.7 KILT Leaderboard

Our performance results on the KILT downstream tasks can be found on the eval.ai leaderboard[9]. We prioritize the performance values reported in the original papers in Table 1 and 2. In cases where

[9] https://eval.ai/web/challenges/challenge-page/689/leaderboard

the original papers do not provide specific values, we rely on the results available on the KILT leaderboard. It is important to note that slight variations in the reported values may occur due to minor differences in the model versions used for evaluation across tasks.

A.8 Metrics

A.8.1 Retrieval

Let us assume that R represents the entire number of retrieved documents, and among these retrieved documents, r is deemed relevant. In this case, R-Precision is the ratio of relevant retrieved documents to the entire number of retrieved documents, i.e., $\frac{r}{R}$. Similarly, Recall@k is calculated as $\frac{w}{n}$, the ratio of relevant retrieved documents to the entire number of actual documents, assuming there are n actual documents and w of these documents were successfully retrieved within a set of k retrieved documents (Petroni et al., 2021).

A.8.2 Context Reranker

Let us consider a classification task with the following definitions: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). Precision is the ratio of true positives to the sum of true and false positives, given by $\frac{\text{TP}}{\text{TP + FP}}$. Similarly, Recall is defined as the ratio of true positives to the sum of true positives and false negatives, denoted as $\frac{\text{TP}}{\text{TP + FN}}$. The F1 score represents a balance between Precision and Recall, computed as the harmonic mean of the two metrics: $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}}$. Accuracy, on the other hand, is calculated as the ratio of the sum of true negatives and true positives to the sum of true negatives, true positives, false positives, and false negatives, given by $\frac{\text{TP + TN}}{\text{TP + TN + FP + FN}}$.

A.8.3 Downstream Performance

For the downstream reading task, we do not perform any post-processing on the gold and predicted outputs for the training and development sets. However, for the blind test sets, KILT applies post-processing techniques such as lowercase conversion, removal of articles, punctuation, and duplicate whitespace to the gold and predicted outputs. KILT maintains that these post-processing steps ensure consistency and fairness in the evaluation process.

A.8.4 KILT scores

As mentioned in 4.2, the KILT score incorporates post-processed Accuracy, EM, ROUGE-L, and F1 scores mentioned in Appendix A.8.3. However, these scores are considered only if the R-Precision for a given query is 1. The KILT scores provide a comprehensive evaluation of the system's performance on the KILT tasks by emphasizing high precision and relevance, in addition to other evaluation metrics.

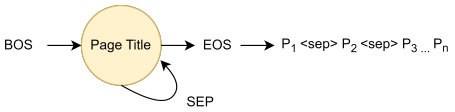
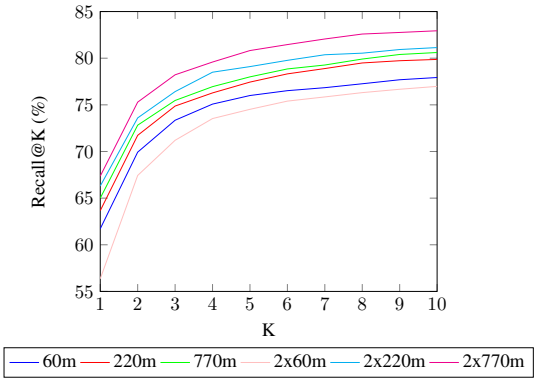


Figure 4: The decoding process in Re3val involves the utilization of DEARDR PTHL state machine decoding. During decoding, each page is conditionally decoded based on the previous page, as there are instances where multiple page titles are mapped to an answer. Furthermore, a query may have various answers, further influencing the decoding process.

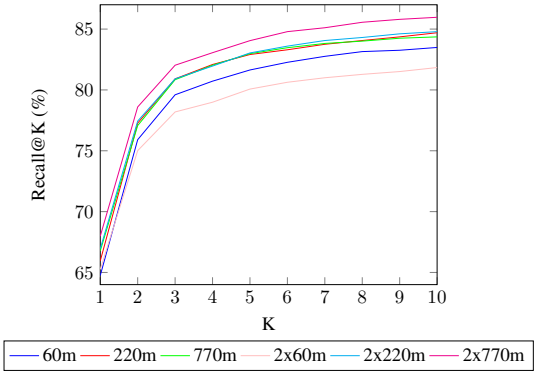
A.9 Recall Curve of the Page Title Reranker

The plots below demonstrate the impact of different numbers of parameters on recall performance at varying levels of documents retrieved. A detailed discussion and analysis of these findings can be found in 5.1 of this paper.

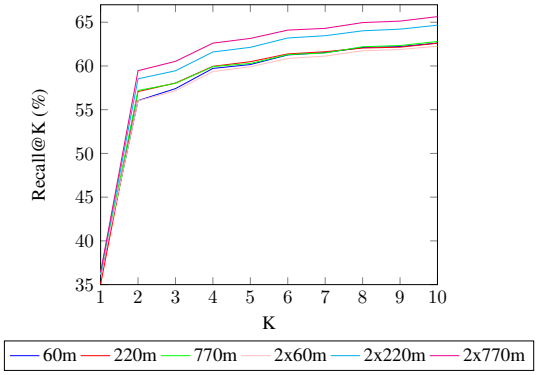
A.9.1 NQ



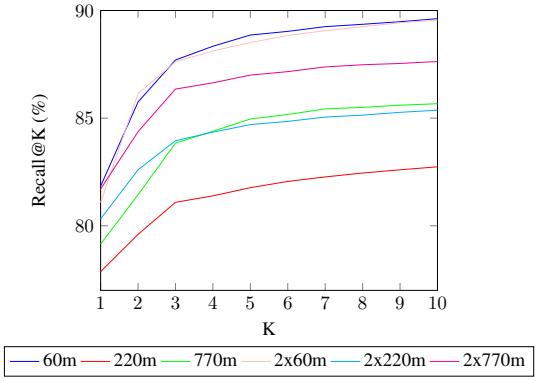
A.9.2 TriviaQA



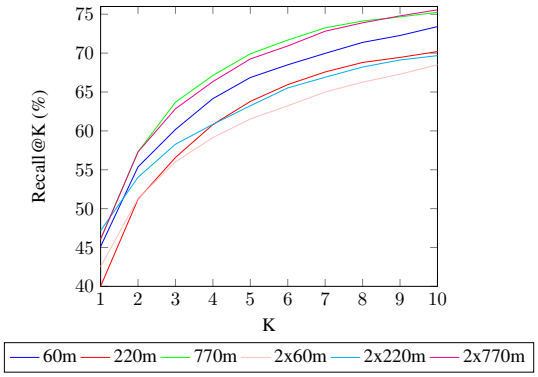
A.9.3 HotpotQA



A.9.4 FEVER



A.9.5 WoW



Question Answering											
NQ				**TQA**				**HoPo**			
PR	**RC**	**F1**	**AC**	**PR**	**RC**	**F1**	**AC**	**PR**	**RC**	**F1**	**AC**
62.04	21.10	31.49	99.12	68.47	32.34	43.93	99.09	79.65	78.76	79.21	99.60

Fact Check.				Dial.			
FEV				**WoW**			
PR	**RC**	**F1**	**AC**	**PR**	**RC**	**F1**	**AC**
76.56	54.35	63.57	99.59	63.45	7.69	13.72	99.56

Table 3: The results of our Context Reranker on the dev sets are presented in terms of Precision (PR), Recall (RC), Accuracy (AC), and F1-Score (F1).

Configuration	**Retrieval**$_L$	**Reranker**$_L$	**Reranker2**	**FiD**
learning rate	5e-4	5e-4	5e-5	1e-4
scheduler	constant w/ warmup	constant w/ warmup	linear	constant
warmup ratio	10%	10%	0	0
eval steps ratio	10%	10%	10%	10%
batch size	46*	10	1200*	32*
max seq length	200*	512	250*	250*
max target length	30	30	50	50
epoch	5*	10*	4	5*
train beam size	1	1	1	1
eval beam size	10	10	1	1
test beam size	5	5	1	1
dropout rate	0.2	0.2	0	0
optimizer	AdamW	AdamW	AdamW	AdamW
gpu	RTX6000	RTX6000	A100	A100
early stopping steps	4	4	4	4

Table 4: The hyperparameter and hardware configurations used in our study are described above. The "Reranker" refers to the page title reranker, while "Reranker2" represents the context reranker. The asterisks (*) denote cases where different values were used for specific tasks. Further information can be found in Tables 5 to 7.

Configuration	**Retrieval**$_S$	**Retrieval**$_B$	**Retrieval**$_L$	**Reranker**$_S$	**Reranker**$_B$	**Reranker**$_L$
batch size	220	160	46	70	35	10
gpu	RTX4000	RTX3090	RTX6000	RTX4000	RTX6000	RTX6000

Table 5: The retrieval and reranker models were configured differently with varying numbers of parameters.

Configuration **Dataset**	**Retrieval**$_S$ WoW	**Retrieval**$_B$ WoW	**Retrieval**$_L$ WoW	**Reranker2** WoW	**FiD** WoW
batch size	110	95	20	600	16
max seq length	512	512	512	500	500

Table 6: The configuration for the Wizard of Wikipedia (WoW) dataset is adjusted to accommodate the longer length of the input.

Configuration	**Retrieval**		**Reranker**			**FiD**
Dataset	FEV	WoW	NQ	FEV	WoW	TQA
epoch	1	1	20	1	1	1

Table 7: Different configurations are utilized for certain datasets, deviating from the settings outlined in 4.

Model	NQ	TQA	HoPo	FEV	WoW
Pre-training					
Re3val	500,000	500,000	500,000	250,000	500,000
GENRE	30,000,000	30,000,000	30,000,000	30,000,000	30,000,000
CorpusBrain	30,000,000	30,000,000	30,000,000	30,000,000	30,000,000
Fine-tuning					
Re3val	48,000	48,000	48,000	48,000	48,000
GENRE	87,372	61,844	88,869	104,966	63,734
CorpusBrain	87,372	61,844	88,869	104,966	63,734

Table 8: The number of datasets utilized for training in our approach is smaller than that employed by other generative retrieval models.

	Question Answering						Fact Check.		Dial.		Average	
Dataset	**NQ**		**TQA**		**HoPo**		**FEV**		**WoW**			
Model	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**
Before Imputation												
Re3val$_S$	59.00	**61.97**	59.69	64.29	54.70	38.18	81.22	85.90	56.90*	71.86*	62.30	**64.44**
Re3val$_B$	64.75	63.05	66.29	71.93	55.76	39.59	81.58	83.27	62.00*	77.50*	66.01	66.67
Re3val$_L$	66.48	65.40	68.55	74.47	59.58	44.21	82.29	85.25	63.32	79.88	67.94	69.13
After Imputation												
Re3val$_S$	**59.63**	60.78	**59.84**	**64.43**	**54.93**	**38.50**	81.22	85.90	56.90*	71.86*	**62.50**	64.29
Re3val$_B$	64.75	63.05	**66.31**	**71.95**	**56.65**	**41.14**	81.58	83.27	62.00*	77.50*	**66.26**	**67.38**
Re3val$_L$	66.48	65.40	68.55	74.47	**59.60**	44.21	**82.37**	85.25	63.32	79.88	**68.06**	69.13

Table 9: The impact of page title imputation using BM-25.

		Question Answering						Fact Check.	Dial.	
Dataset	**|P|**	**NQ**		**TQA**		**HoPo**		**FEV**	**WoW**	
Model		EM	F1	EM	F1	EM	F1	AC	RL	F1
Few-shot (48k)										
Re3val	5	39.06	48.58	40.49	50.54	35.13	45.60	88.25	17.06	17.49
Re3val$_I$	5	**41.50**	51.02	40.98	51.15	36.27	**47.15**	89.83	17.68	17.87
Re3val	10	40.36	51.15	42.84	53.29	35.09	46.02	88.42	17.22	17.56
Re3val$_I$	10	41.35	**51.84**	**43.35**	**53.74**	**36.30**	46.93	**90.09**	**17.83**	**17.90**

Table 10: The best scores achieved on the dev sets when fine-tuning FiD are presented in the table above. The values highlighted in **bold** indicate the best scores, while those underlined indicate the second-best scores. The notation *I* represents the *Imputation* of DPR contexts for missing gold contexts.

Dataset	|P|	Question Answering NQ		TQA		HoPo		Fact Check. FEV	Dial. WoW	
Model		EM	F1	EM	F1	EM	F1	AC	RL	F1
Pre-training (48k)										
Re3val	5	44.88	52.86	62.24	67.17	31.78	40.78	86.30	14.53	15.89
Re3val$_I$	5	48.75	56.58	66.23	70.65	33.90	43.49	89.43	14.74	16.36
Full Fine-tuning										
SEAL	100	**53.74**	**62.24**	70.86	**77.29**	**40.46**	**51.44**	89.54	16.65	18.34
RAG	5	44.39	52.35	**71.27**	75.88	26.97	36.03	86.31	11.57	13.11
KGI	5	45.22	53.38	60.99	66.55	-	-	85.58	16.36	18.57
DPR + BART	5	39.75	48.43	59.60	66.53	31.77	41.56	86.32	13.27	15.12
Few-shot (48k)										
Re3val	5	47.92	56.46	64.39	69.14	35.39	45.04	87.36	16.75	19.03
Re3val	10	49.79	58.94	66.57	71.42	35.73	45.48	87.15	16.92	18.93
Re3val$_I$	5	49.58	57.75	65.06	69.96	36.45	46.66	89.27	**17.10**	19.06
Re3val$_I$	10	48.68	57.37	65.87	70.49	36.52	46.89	**89.59**	17.06	**19.16**

Table 11: Reader scores of test sets on the KILT Leaderboard. The **bolded** are the best and the underlined are the second best. *I* indicates the *Imputation* of DPR contexts for missing gold contexts. Note that the reader scores are not final scores as final scores are the KILT scores which award reader scores if R-Precision is 1.

Dataset			Question Answering NQ		TQA		HoPo		Fact Check. FEV		Dial. WoW	
Model	**|P|**	**Stage**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**	**R-P**	**R@5**
Re3val	60m	*Z*	26.40	35.35	45.62	59.38	52.95	45.91	77.70	84.93	46.40	58.91
Re3val	60m	*Z*, *P*	27.42	36.02	46.05	58.95	52.67	45.94	78.49	85.92	44.27	56.81
Re3val	60m	*F*	45.40	60.49	59.49	71.99	51.06	49.45	81.74	87.73	**48.10**	**67.62**
Re3val	60m	*F*, *P*	47.59	62.18	60.68	73.00	50.45	49.59	81.90	87.60	46.23	65.88
Re3val	60m	*R*	61.72	76.00	64.75	**81.64**	56.79	60.16	**84.79**	**88.86**	45.12	66.86
Re3val	60m	*R*, *P*	**62.39**	75.36	63.78	81.36	**57.39**	**60.32**	**84.79**	88.07	43.98	67.13
Re3val	60m,60m	*R*	56.36	74.52	**65.25**	80.07	57.04	59.91	83.87	88.51	42.53	61.53
Re3val	60m,60m	*R*, *P*	61.37	**76.67**	64.43	80.29	56.72	59.73	82.94	87.93	36.97	58.32
Re3val	220m	*Z*	32.78	45.93	47.02	62.72	52.29	46.78	72.27	85.98	49.84	60.31
Re3val	220m	*Z*, *P*	35.78	47.97	42.40	60.59	54.13	47.64	77.25	86.81	49.18	61.85
Re3val	220m	*F*	54.74	69.05	61.90	77.87	50.69	51.97	79.15	82.58	52.00	71.77
Re3val	220m	*F*, *P*	54.35	68.56	61.78	78.52	50.43	51.88	78.74	81.95	**52.72**	**72.10**
Re3val	220m	*R*	63.66	77.44	65.95	82.91	57.54	60.49	79.82	81.77	40.01	63.79
Re3val	220m	*R*, *P*	64.22	76.35	65.80	82.87	57.69	60.39	79.86	82.52	39.06	62.41
Re3val	220m,220m	*R*	**66.30**	**79.10**	**66.95**	**83.04**	**58.85**	**62.13**	82.39	**84.70**	47.18	63.23
Re3val	220m,220m	*R*, *P*	65.67	78.43	64.51	80.71	58.73	61.82	**82.84**	84.59	39.06	62.38
Re3val	770m	*Z*	32.11	47.83	43.37	61.19	48.10	46.33	78.73	83.77	49.67	65.55
Re3val	770m	*Z*, *P*	33.84	49.77	44.95	63.22	46.24	44.90	81.08	**87.94**	50.36	65.19
Re3val	770m	*F*	55.97	71.24	64.06	79.92	50.39	51.85	80.46	82.97	**55.34**	**74.89**
Re3val	770m	*F*, *P*	57.00	71.23	63.61	79.79	50.62	52.27	79.40	82.40	53.90	74.36
Re3val	770m	*R*	65.00	78.00	66.77	82.98	57.66	60.29	81.64	84.96	46.07	69.91
Re3val	770m	*R*, *P*	64.65	78.22	67.25	81.82	57.95	60.48	81.26	84.74	38.47	62.38
Re3val	770m,770m	*R*	**67.36**	**80.82**	**67.98**	**84.05**	59.75	63.15	**84.68**	87.00	46.07	69.25
Re3val	770m,770m	*R*, *P*	63.80	77.79	65.05	79.79	**59.76**	**63.26**	81.43	82.77	46.73	69.68

Table 12: The performance of the development sets is evaluated at each stage of the training, considering different numbers of parameters. The stages include zero-shot retrieval (Z), few-shot retrieval (F), reranking (R), and reinforcement (P). The parameter counts $|P|$ represent the total parameters used to train the retrieval and reranker models. The comma (,) in $|P|$ indicates that the retrieval and reranker were initialized separately. In contrast, the absence of a comma (,) signifies that the reinforced few-shot retrieval was fine-tuned with the reranker's input and output.