Stochastic RAG: End-to-End Retrieval-Augmented Generation through Expected Utility Maximization

Hamed Zamani
University of Massachusetts Amherst
Amherst, MA, United States
zamani@cs.umass.edu

Michael Bendersky
Google
Mountain View, CA, United States
bemike@google.com

ABSTRACT

This paper introduces Stochastic RAG–a novel approach for end-to-end optimization of retrieval-augmented generation (RAG) models that relaxes the simplifying assumptions of marginalization and document independence, made in most prior work. Stochastic RAG casts the retrieval process in RAG as a stochastic sampling without replacement process. Through this formulation, we employ straight-through Gumbel-top-k that provides a differentiable approximation for sampling without replacement and enables effective end-to-end optimization for RAG. We conduct extensive experiments on seven diverse datasets on a wide range of tasks, from open-domain question answering to fact verification to slot-filling for relation extraction and to dialogue systems. By applying this optimization method to a recent and effective RAG model, we advance state-of-the-art results on six out of seven datasets.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**;
• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Retrieval augmentation; retrieval-enhanced machine learning; ranking optimization

ACM Reference Format:
Hamed Zamani and Michael Bendersky. 2024. Stochastic RAG: End-to-End Retrieval-Augmented Generation through Expected Utility Maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3626772.3657923

1 INTRODUCTION

Most machine learning systems, including large generative models, are self-contained systems, with both knowledge and reasoning encoded in model parameters. However, these models do not work effectively for tasks that require knowledge grounding [46], especially in case of non-stationary data where new information is actively being produced [47, 52]. As suggested by Zamani et al. [52], this issue can be addressed when machine learning systems are being *enhanced with the capability of retrieving stored content.* For example, in retrieval-augmented generation (RAG), as a special case of retrieval-enhanced machine learning (REML) [52], systems consume the responses provided by one or more retrieval models for the purpose of (text) generation [21, 22]. RAG models demonstrate substantial promise across various applications, including open-domain question answering [16, 21, 53], fact verification [44], dialogue systems [5, 42, 48], and personalized generation [36, 37].

Many prior studies on RAG use off-the-shelf retrieval models. For instance, Nakano et al. [25] used APIs from a commercial search engine for text generation. Glass et al. [9], on the other hand, used a term matching retrieval model. Neural ranking models trained based on human annotated data have also been used in the literature [12, 21]. There also exist methods that only optimize the retrieval model and keep the language model parameters frozen [40]. A research direction in this area argues that optimizing retrieval models in RAG should depend on the downstream language model that consumes the retrieval results. This is also motivated by the findings presented by Salemi and Zamani [38] on evaluating retrieval quality in RAG systems. There exist solutions based on knowledge distillation [13] or end-to-end optimization based on some simplifying assumptions [35]. One of these assumptions is *marginalization via top k approximation* [10, 21]. In more details, they first retrieve the top k documents using off-the-shelf retrieval models, e.g., BM25 [34], and optimize retrieval models by *re-scoring* them, i.e., re-ranking, and feeding the documents to the downstream language model one-by-one independently [21]. This is far from reality as RAG models often consume multiple documents.

This paper introduces Expected Utility Maximization for RAG–a novel framework for end-to-end RAG optimization by relaxing these simplifying assumptions. This approach takes a utility function, which can be any arbitrary evaluation metric for the downstream generation task, such as exact match, BLEU [26], and ROUGE [23]. A major challenge in end-to-end optimization of RAG systems is that ranking and top k selection is a non-differentiable process. Hence, this prevents us from using gradient descent-based methods for optimization. We address this issue by casting retrieval as a *sampling without replacement* process from the retrieval score distribution, which is approximated using the straight-through Gumbel-top-k approach. This stochastic approach—called Stochastic RAG—adds a Gumbel noise to the unnormalized retrieval scores and uses softmax to approximate argmax [17, 18].

Stochastic RAG can be applied to any RAG application. We evaluate our models using seven datasets from a wide range of applications, ranging from open-domain question answering to fact verification to slot-filling for relation extraction as well as dialogue systems. We apply our optimization method to FiD-Light [12], which

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657923

is the best performing system on six out of these seven datasets, according to the knowledge-intensive language tasks (KILT) leaderboard as of Feb. 1, 2024.[1] Our results demonstrate significant improvements on all these datasets.

2 EXPECTED UTILITY MAXIMIZATION FOR STOCHASTIC RAG

Each RAG system consists of two main components: a text generation model G_θ parameterized by θ and a retrieval model R_ϕ parameterized by ϕ that retrieves documents from a large document collection C. The text generation model consumes the retrieval results returned by the retrieval model. End-to-end optimization of RAG systems is challenging. This is mainly because retrieving top k documents and feeding them to the generation model is not a differentiable process [52], thus one cannot simply employ gradient-based optimization algorithms for end-to-end optimization of these models. In this section, we introduce stochastic expected utility maximization for end-to-end optimization of retrieval-augmented models.

Let $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ be a training set containing n pairs of x_i (an input text) and y_i (the ground truth output text). Let U denote a utility function that takes the output generated by the RAG system $\hat{y}$ and the ground truth output y and generates a scalar value. The utility function can be any arbitrary metric, including but is not limited to, exact match, term overlap F1, BLEU, and ROUGE. We assume (1) the higher the utility value, the better, (2) the utility function is bounded within the $[0, 1]$ range, and (3) $U(y, y) = 1$. We define RAG Expected Utility as follows:

$$\text{RAG Expected Utility} = \frac{1}{n} \sum_{(x,y)\in T} \sum_{\hat{y}\in\mathcal{Y}} U(y, \hat{y}) p(\hat{y}|x; G_\theta, R_\phi) \tag{1}$$

where $\mathcal{Y}$ the output space, i.e., all possible output texts. In some models, the output space is limited, for instance in fact verification, the output space is often binary: the given candidate fact is often true or false. In other situations, such as free-form text generation, the output space is unlimited. To make sure that expected utility calculation is tractable, we would need to approximate the above equation by sampling from the unlimited space $\mathcal{Y}$. We will explain how such samples can be obtained at the end of this section.

The probability of generating any given output $\hat{y}$ in a RAG system can be modeled as:

$$\begin{aligned} p(\hat{y}|x; G_\theta, R_\phi) &= \sum_{\mathbf{d}\in\pi_k(C)} p(\hat{y}, \mathbf{d}|x; G_\theta, R_\phi) \\ &= \sum_{\mathbf{d}\in\pi_k(C)} p(\hat{y}|x, \mathbf{d}; G_\theta) p(\mathbf{d}|x; G_\theta, R_\phi) \\ &= \sum_{\mathbf{d}\in\pi_k(C)} p(\hat{y}|x, \mathbf{d}; G_\theta) p(\mathbf{d}|x; R_\phi) \end{aligned} \tag{2}$$

where $\pi_k(C)$ denotes all permutations of k documents being selected from the retrieval collection C. The first step in the above equation is obtained using the law of total probability, the second step is obtained using the chain rule, and the third step is obtained due to the fact that the probability of a result list $\mathbf{d}$ being retrieved is independent of the text generation model G_θ.

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

Note that considering all permutations in $\pi_k(C)$ is expensive and impractical for large collections, thus we can compute an approximation of this equation. We do such approximation through a stochastic process. We rewrite Equation (2) as follows:

$$p(\hat{y}|x; G_\theta, R_\phi) = \mathbb{E}_{\mathbf{d}\sim p(\mathbf{d}|x;R_\phi)} \left[p(\hat{y}|x, \mathbf{d}; G_\theta) \right] \tag{3}$$

where $|\mathbf{d}| = k$. Inspired by the seq2seq models [43], we compute $p(\hat{y}|x, \mathbf{d}; G_\theta)$—the component in Equation (2)—as follows:

$$\begin{aligned} p(\hat{y}|x, \mathbf{d}; G_\theta) &= \prod_{i=1}^{|\hat{y}|} p(\hat{y}_i|\hat{y}_{<i}, x, \mathbf{d}; G_\theta) \\ &= \exp\left(\sum_{i=1}^{|\hat{y}|} \log p(\hat{y}_i|\hat{y}_{<i}, x, \mathbf{d}; G_\theta) \right) \end{aligned} \tag{4}$$

where $\hat{y}_i$ denotes the i^{th} token in $\hat{y}$ and $\hat{y}_{<i}$ denotes all tokens $\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{i-1}$.

The next step is to estimate $p(\mathbf{d}|x; R_\phi)$ in Equation (3), which represents the probability of retrieving the result list $\mathbf{d}$ in response to input x using the retrieval model R_ϕ. Most retrieval models score each query-document pair independently and then sort them with respect to their relevance score in descending order. Therefore, the probability of a document list being produced by R_ϕ can be modeled as a *sampling without replacement* process. In other words, assume that the retrieval model R_ϕ produces a retrieval score $s^\phi_{xd} \in \mathbb{R}$ for any document $d \in C$. Sampling without replacement probability of a document list is then computed as:

$$p(\mathbf{d}|x; R_\phi) = \prod_{i=1}^{|\mathbf{d}|} \frac{p(d_i|x; R_\phi)}{1 - \sum_{j=1}^{i-1} p(d_j|x; R_\phi)} \tag{5}$$

where document-level probabilities $p(d_i|x; R_\phi)$ can be computed using the softmax operation:

$$p(d_i|x; R_\phi) = \frac{\exp(s^\phi_{xd_i})}{\sum_{d\in C} \exp(s^\phi_{xd})} \tag{6}$$

This iterative process of document sampling is non-differentiable, and thus cannot be simply used in gradient descent-based optimization approaches. To address both of these problems, Kool et al. [17, 18] recently introduced Ancestral Gumbel-Top-k sampling. This approach creates a tree over all items in the sampling set and extends the Gumbel-Softmax sampling approach [24] to sampling without replacement. According to [17], independently perturbing each individual document score with Gumbel noise and picking the top k documents with the largest perturbed values will generate a valid sample from the Plackett-Luce distribution. Gumbel perturbation itself can be done efficiently by simply drawing a sample $U \sim \text{Uniform}(0, 1)$, as $\text{Gumbel}(0, \beta) \sim -\beta \log(-\log(U))$ [24].

$$\tilde{p}(d_i|\phi, \theta) = \frac{\exp(s^\phi_{xd_i} + G_{d_i})}{\sum_{d\in C} \exp(s^\phi_{xd} + G_d)} \tag{7}$$

where G_d denotes the gumbel noise added for scoring document d.

We use *straight-through gumbel-top-k*, in which the top k elements are selected from the above distribution using the arg max operation in the forward path, however, the softmax distribution is

used in the backward path for computing the gradients. For more information on straight-through gumbel-softmax, refer to [14, 28]. Gumbel-top-k has been used in IR systems too. For instance, Zamani et al. [51] used the gumbel-top-k trick to optimize re-ranking models conditioned on the first stage retrieval models.

Selecting $\mathcal{Y}$. In Equation (1), $\mathcal{Y}$ denotes the output space, which can be unlimited for free-form text generation tasks, hence computationally intractable. In such cases, we need to estimate RAG Expected Utility by sampling from the output space. A uniformly random sample can give us an unbiased estimation, however, most random samples are completely unrelated to the input, so they can be easily discriminated from the ground truth output. Inspired by work on hard negative sampling for training ranking models [31, 49], at every $N = 10,000$ training steps, we run the RAG model that is being trained on the training inputs that will be used in the next N steps and use beam search to return 100 most probable outputs. We randomly sample $m = 10$ of these outputs to form $\mathcal{Y}$. We then made sure that for every pair (x, y) in the training set for the next N steps, y is included in $\mathcal{Y}$, otherwise we randomly replace one of the sampled outputs in $\mathcal{Y}$ with y. The reason for doing this is to make sure that our sample contains the ground truth output, ensuring that the model learns to produce higher probability for the ground truth output. Preparing $\mathcal{Y}$ for the next N training steps would also enable us to pre-compute utility values $U(y, \hat{y}) : \forall \hat{y} \in \mathcal{Y}$, ensuring an efficient optimization process for RAG Expected Utility Maximization (see Equation (1)).

3 EXPERIMENTS

3.1 Data

We use the Natural Questions (NQ) [19], TriviaQA [15], HotpotQA [50], FEVER [45], T-REx [7], zsRE [20], and Wizard of Wikipedia (WoW) [6] datasets from the KILT [29] benchmark. Due to the unavailability of ground truth labels for test set, our experiments are conducted on the publicly accessible validation sets. As the retrieval corpus, we employ the Wikipedia dump provided with the KILT benchmark[2] and adhere to the preprocessing steps outlined by Karpukhin et al. [16], where each document is segmented into passages, each constrained to a maximum length of 100 words. The concatenation of the article title and passage text is used as a document. Note that the KILT benchmark furnishes document-level relevance labels (called Provenance) for its datasets, and these are employed for evaluating retrieval performance. In line with our preprocessing method outlined in this paper, we define all passages within a positive document as positive passages for our evaluation.

For evaluating our models, we follow the standard KILT evaluation setup [29] by focusing on KILT-score metrics. KILT-scores combine R-Precision (RP) obtained by the retrieval results and the quality of the generated output text that is evaluated using any arbitrary metric M (such as EM, Accuracy, or F1). For a query set Q, KILT-scores are computed as follows:

$$\text{KILT-M} = \frac{1}{|Q|} \sum_{q \in Q} \{RP(\mathbf{p}, \mathbf{d}) == 1\} * M(y, \hat{y}) \tag{8}$$

[2]Retrieval corpus: https://dl.fbaipublicfiles.com/ur/wikipedia_split/psgs_w100.tsv.gz

where $\mathbf{d}$ is the retrieval results produced by the retrieval model, $\mathbf{p}$ is the provenance label set provided by KILT, y is the ground truth output, and $\hat{y}$ is the generated text. Note that there is only one provenance label per query in most KILT datasets. FEVER and HotPotQA are the only exceptions. 12% of queries are associated with more than one supporting document in FEVER and all queries in HotPotQA (which focuses on multi-hop question answering) are associated with two documents. KILT-scores only evaluates the generated text if R-Precision is 1. This means that it does not solely focus on the quality of the generated text, but also makes sure that relevant supporting documents are provided. We adopt the metrics recommended by the KILT benchmark, namely Exact Match (KILT-EM) for NQ, TriviaQA, and HotpotQA, Accuracy (KILT-AC) for FEVER, and F1-score (KILT-F1) for the WoW dataset.

3.2 Experimental Setup

We apply the proposed optimization framework to a state-of-the-art RAG model on the KILT benchmark (i.e., FiD-Light, according to the KILT leaderboard) [29]. Therefore, we follow the experimental setup of Hofstätter et al. [12] for FiD-Light. That means we used multi-task relevance sampled training set from the authors earlier work in [11] and trained a dense retrieval model, which is pre-trained on the MSMARCO passage retrieval data [2]. Given that the datasets in our experiments focuses on relatively short-text generation tasks, and since all passages are less than or equal to 100 tokens, we set the input token limit for both query and passage combined at 384 tokens and for the output at 64 tokens. For training, we use a batch size of 128 with up to 40 retrieved passages, and a learning rate of 10^{-3} with the Adafactor optimizer [39]. We trained our models for 50,000 steps. We cut the learning rate by half for the large language models (i.e., T5-XL). During decoding, we use beam search with a beam size of 4. All our experiments are based on the T5X framework [33] on TPUs using T5v1.1 as the language model backbone [32]. For each dataset, we use the official KILT-score metric as the utility function for optimization (Equation (1)).

3.3 Results

To evaluate the effectiveness of the RAG Expected Utility Maximization framework, we compare our model with the best performing entries in the KILT leaderboard (as of February 1, 2024) according to the official KILT-score metrics. These methods use a wide range of techniques to address these issues including dense retrieval methods followed by BART or T5 for generation, generative retrieval models, retrieval and reranking models, pre-trained large language models without augmentation, etc. These methods and their corresponding references are listed in Table 1. For the sake of space, we do not list their underlying methods here. The performance of these methods is obtained from the KILT leaderboard. We use FiD-Light as the main baseline in this paper, as it produces state-of-the-art results on six out of seven datasets and the proposed optimization method is applied to FiD-Light. FiD-Light is a simple extension of the Fusion-in-Decoder architecture that generates the document identifier of relevant documents in addition to the output text and uses then at inference for re-ranking the input result list. According to the results presented in Table 1, employing stochastic expected

Table 1: Comparing our models with top performing entries in the KILT leaderboard according to KILT-scores, as of February 1, 2024. The results are reported on the blind KILT test sets.

Model	Open Domain QA			Fact	Slot Filling		Dialog
	NQ	HotpotQA	TriviaQA	FEVER	T-REx	zsRE	WOW
	KILT-EM	*KILT-EM*	*KILT-EM*	*KILT-AC*	*KILT-AC*	*KILT-AC*	*KILT-F1*
RAG [21]	32.7	3.2	38.1	53.5	23.1	36.8	8.8
DPR + FiD [30]	35.3	11.7	45.6	65.7	64.6	67.2	7.6
KGI [8]	36.4	–	42.9	64.4	69.1	72.3	11.8
Re2G [10]	43.6	–	57.9	78.5	75.8	–	12.9
Hindsight [27]	–	–	–	–	–	–	13.4
SEAL + FiD [4]	38.8	18.1	50.6	71.3	60.1	73.2	11.6
Re3val [41]	39.5	24.2	51.3	73.0	–	–	13.5
GripRank [1]	43.6	–	58.1	–	–	79.9	**14.7**
PLATO [3]	–	–	–	–	–	–	13.6
FiD-Light (T5-Base, $k = 64$)	45.6	25.6	57.6	80.6	76.0	81.1	11.9
FiD-Light (T5-XL, $k = 8$)	51.1	29.2	63.7	84.5	76.3	84.0	13.1
Stochastic RAG with FiD-Light (T5-Base, $k = 64$)	46.2	27.3	59.7	81.3	76.9	82.8	12.8
Stochastic RAG with FiD-Light (T5-XL, $k = 8$)	**53.0**	**31.1**	**64.7**	**84.8**	**78.3**	**87.0**	14.2

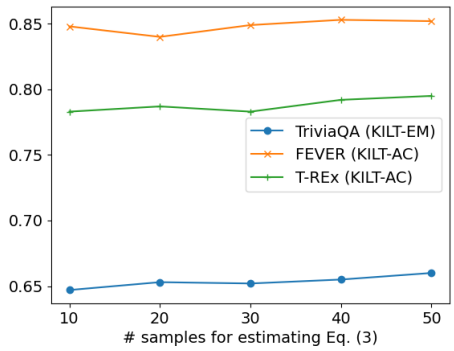

Figure 1: Sensitivity of Stochastic RAG with FiD-Light XL to the number of samples for estimating Equation (3).

utility maximization leads to improvements in all datasets. Comparing against state-of-the-art baselines from the KILT leaderboard, our approach presents the best performing result in all datasets except for Wizard of Wikipedia, where only one method, named GripRank, performs slightly better than our best performing system. Note that in another dataset (i.e., zsRE), our methods outperform GripRank by a large margin.

The last two rows in Table 1 present the results for the same model with different sizes for the downstream language model. T5-Base contains 220 million parameters, while T5-XL is a language model with 3 billion parameters. We observe that both model sizes benefit from applying stochastic expected utility maximization. As expected, the larger model exhibits a better performance. That said, the performance difference between the Base and XL size models is not consistent across datasets. For instance, we observe substantial relative improvements on Natural Questions (i.e., 14.5%), while improvements on T-REx are smaller (i.e., 1.8%).

To provide a deeper analysis of the Stochastic RAG performance, we vary the number of samples we take for estimating Equation (3). For the sake of visualization, we only present the results for a QA, a fact verification, and a slot-filling dataset in Figure 1. We observe that the model is robust with respect to the different number of samples. That said, sometimes we observe slight improvement as we increase the sample size (e.g., on TriviaQA).

4 CONCLUSIONS AND FUTURE WORK

This paper presented a novel optimization framework for end-to-end optimization of retrieval-augmented generation models. The framework maximizes stochastic expected utility, where the utility can be any arbitrary evaluation metric appropriate for the downstream generation task. Without loss of generality, we applied this optimization approach to FiD-Light as an effective RAG model and observed substantial improvements on seven diverse datasets from the KILT benchmark. We demonstrate that the proposed approach advances state-of-the-art results on six out of seven datasets on the blind test sets provided by the benchmark. Our results suggest that language models of different sizes (220M parameters and 3B parameters) benefit from such end-to-end optimization.

This work solely focuses on relatively short text generation. In the future, we aim at studying the impact of Stochastic RAG on long text generation and exploring various utility functions that can be defined in RAG optimization. Furthermore, the stochastic nature of Stochastic RAG can be used to increase the diversity of generated outputs in RAG systems. This is quite important in scenarios where multiple outputs are generated by RAG systems for collecting human feedback.

ACKNOWLEDGMENTS

We thank the reviewers for their invaluable feedback. This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant number 2143434, in part by the Office of Naval Research contract number N000142212688, and in part by an award from Google. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Jiaqi Bai, Hongcheng Guo, Jiaheng Liu, Jian Yang, Xinnian Liang, Zhao Yan, and Zhoujun Li. 2023. GripRank: Bridging the Gap between Retrieval and Generation via the Generative Knowledge Improved Passage Ranking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 36–46. https://doi.org/10.1145/3583780.3614901

[2] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[3] Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhihua Wu, Zhen Guo, Hua Lu, Xinxian Huang, Xin Tian, Xinchao Xu, Yingzhan Lin, and Zheng-Yu Niu. 2022. PLATO-XL: Exploring the Large-scale Pre-training of Dialogue Generation. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang (Eds.). Association for Computational Linguistics, Online only, 107–118. https://aclanthology.org/2022.findings-aacl.10

[4] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive Search Engines: Generating Substrings as Document Identifiers. *arXiv preprint arXiv:2204.10628* (2022).

[5] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241* (2018).

[6] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1l73iRqKm

[7] Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[8] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, and Alfio Gliozzo. 2021. Robust retrieval augmented generation for zero-shot slot filling. *arXiv preprint arXiv:2108.13934* (2021).

[9] 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*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 2701–2715. https://doi.org/10.18653/v1/2022.naacl-main.194

[10] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, Rerank, Generate. *arXiv preprint arXiv:2207.06300* (2022).

[11] Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2022. Multi-Task Retrieval-Augmented Text Generation with Relevance Sampling. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

[12] Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. FiD-Light: Efficient and Effective Retrieval-Augmented Text Generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 1437–1447. https://doi.org/10.1145/3539618.3591687

[13] Gautier Izacard and Edouard Grave. 2020. Distilling Knowledge from Reader to Retriever for Question Answering. https://arxiv.org/abs/2012.04584

[14] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations (ICLR '17)*.

[15] 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)*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, Vancouver, Canada, 1601–1611. https://doi.org/10.18653/v1/P17-1147

[16] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[17] Wouter Kool, Herke Van Hoof, and Max Welling. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*. PMLR, 3499–3508.

[18] Wouter Kool, Herke van Hoof, and Max Welling. 2020. Ancestral Gumbel-Top-k Sampling for Sampling Without Replacement. *J. Mach. Learn. Res.* 21 (2020), 47–1.

[19] 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 (2019), 452–466. https://doi.org/10.1162/tacl_a_00276

[20] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115* (2017).

[21] 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 (2020), 9459–9474.

[22] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A Survey on Retrieval-Augmented Text Generation. arXiv:2202.01110 [cs.CL]

[23] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[24] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=S1jE5L5gl

[25] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332 [cs.CL]

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. https://doi.org/10.3115/1073083.1073135

[27] Ashwin Paranjape, Omar Khattab, Christopher Potts, Matei Zaharia, and Christopher D Manning. 2021. Hindsight: Posterior-guided training of retrievers for improved open-ended generation. *arXiv preprint arXiv:2110.07752* (2021).

[28] Max B Paulus, Chris J. Maddison, and Andreas Krause. 2021. Rao-Blackwellizing the Straight-Through Gumbel-Softmax Gradient Estimator. In *International Conference on Learning Representations (ICLR '21)*.

[29] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick S. H. 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, NAACL-HLT 2021, Online, June 6-11, 2021*.

[30] Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, et al. 2021. The Web Is Your Oyster–Knowledge-Intensive NLP against a Very Large Web Corpus. *arXiv preprint arXiv:2112.09924* (2021).

[31] Prafull Prakash, Julian Killingback, and Hamed Zamani. 2021. Learning Robust Dense Retrieval Models from Incomplete Relevance Labels. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1728–1732. https://doi.org/10.1145/3404835.3463106

[32] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21 (2020), 1–67.

[33] Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2022. Scaling Up Models and Data with `t5x` and `seqio`. *arXiv preprint arXiv:2203.17189* (2022).

[34] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Text Retrieval Conference*. https://api.semanticscholar.org/CorpusID:3946054

[35] Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-End Training of Neural Retrievers 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)*. Association for Computational Linguistics, Online, 6648–6662. https://doi.org/10.18653/v1/2021.acl-long.519

[36] Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024. Optimization Methods for Personalizing Large Language Models through Retrieval Augmentation. In *Proceedings of the 47th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington, DC, USA) *(SIGIR '24)*. (to appear).

[37] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When Large Language Models Meet Personalization. arXiv:2304.11406 [cs.CL]

[38] Alireza Salemi and Hamed Zamani. 2024. Evaluating Retrieval Quality in Retrieval-Augmented Generation. In *Proceedings of the 47th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington, DC, USA) *(SIGIR '24)*. (to appear).

[39] Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*. PMLR, 4596–4604.

[40] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. REPLUG: Retrieval-Augmented Black-Box Language Models. arXiv:2301.12652 [cs.CL]

[41] EuiYul Song, Sangryul Kim, Haeju Lee, Joonkee Kim, and James Thorne. 2024. Re3val: Reinforced and Reranked Generative Retrieval. arXiv:2401.16979 [cs.IR]

[42] Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. 2018. An Ensemble of Retrieval-Based and Generation-Based Human-Computer Conversation Systems. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 4382–4388. https://doi.org/10.24963/ijcai.2018/609

[43] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS '14, Vol. 27)*. Curran Associates, Inc.

[44] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355* (2018).

[45] 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)*, Marilyn Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, New Orleans, Louisiana, 809–819. https://doi.org/10.18653/v1/N18-1074

[46] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the Planning Abilities of Large Language Models - A Critical Investigation. *arXiv 2305.15771* (2023).

[47] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2023. FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation. In *arXiv*.

[48] Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and Refine: Improved Sequence Generation Models For Dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*. Association for Computational Linguistics, Brussels, Belgium, 87–92. https://doi.org/10.18653/v1/W18-5713

[49] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations (ICLR '21)*.

[50] 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*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 2369–2380. https://doi.org/10.18653/v1/D18-1259

[51] Hamed Zamani, Michael Bendersky, Donald Metzler, Honglei Zhuang, and Xuanhui Wang. 2022. Stochastic Retrieval-Conditioned Reranking. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval* (Madrid, Spain) *(ICTIR '22)*. Association for Computing Machinery, New York, NY, USA, 81–91. https://doi.org/10.1145/3539813.3545141

[52] Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. 2022. Retrieval-Enhanced Machine Learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2875–2886. https://doi.org/10.1145/3477495.3531722

[53] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. arXiv:2101.00774 [cs.AI]