Reinforced Queries using Pre-trained Language Models in Sparse Retrieval

Anonymous Author(s)*

ABSTRACT

Despite the success of dense retrieval, sparse retrieval methods still show potential in interpretability and generalizability. However, query-document term mismatch in sparse retrieval persists, rendering it infeasible for many practical applications. To remedy this, we introduce a novel query expansion approach, denoted as *QSparse*. QSparse generates expanded terms by pre-trained language models trained by reinforcement learning and then uses a sparse retrieval method to retrieve documents. A thorough experimental evaluation on three datasets from disparate domains (SCIFACT, Natural Questions (NQ), and MS-MARCO passage) shows that QSparse enriches the original query and significantly improves sparse retrieval. Furthermore, QSparse, when combined with dense retrieval, achieves an 8% improvement in NDCG@10 for SCIFACT and a 2% increase in recall for NQ, compared to the original dense retrieval. These results highlight that QSparse leverages the benefits of both sparse retrieval and dense retrieval to address mismatch issues. Our solution and code are publicly available on GitHub[1].

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Query expansion, Pre-trained language model, Reinforcement learning

ACM Reference Format:

Anonymous Author(s). 2018. Reinforced Queries using Pre-trained Language Models in Sparse Retrieval. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The fundamental mechanism underlying sparse retrieval methods involve matching the words in a given query with those in relevant documents. This mechanism is prone to query-document term mismatch, resulting in the omission of relevant documents [25]. *Query expansion* is a widely used technique to address mismatch issues in sparse retrieval. Traditional query expansion approaches such as *Pseudo Relevance Feedback* (PRF) rely on retrieving pseudo-relevant documents to expand queries [14, 15, 19, 22, 40]. Recent advances in query expansion techniques tend to rely on the generative capabilities and inherent knowledge of pre-trained language models (PLMs) [21, 40]. Among them, large language models (LLMs) like GPT from OpenAI [5] have demonstrated impressive abilities; however, the expanded terms generated by PLMs may introduce irrelevant information [25].

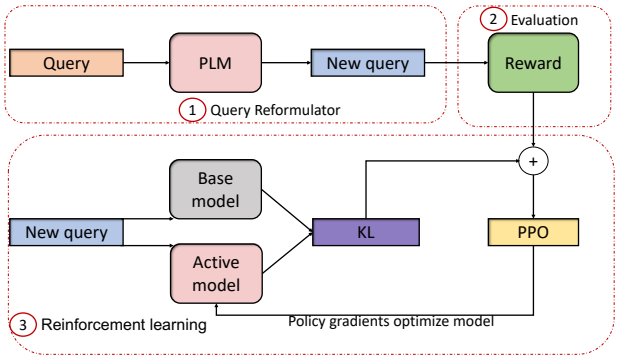

Figure 1: High-level overview of QSparse. PPO refers to the RL optimization algorithm called Proximal Policy Optimization (PPO) [33].

In this paper, we propose the use of PLMs fine-tuned by reinforcement learning (RL) to aid in query expansion. Specifically, we use a PLM, GPT2 [29], fine-tuned using the Reinforcement Learning from Human Feedback (RLHF) paradigm [3, 23, 26, 34, 41]. RL has been successfully applied to enhance PLMs in various tasks, including question answering [6], conversational question answering [7, 10, 39], and conversational query reformulation [17]. We argue that RL can reduce irrelevant information in query expansion induced by PLMs, helping PLMs address matching issues in sparse retrieval.

We name our approach as QSparse, and its three major steps are shown in Figure 1: **(1)** During query expansion, a PLM generates expanded terms, which are then concatenated with original queries (see Section 3.1). **(2)** At the reward stage, new queries are evaluated with a reward function to yield a scalar value. We use evaluation metrics, e.g., NDCG@10, as the reward function (see Section 3.2). **(3)** During RL optimization, there is an *active model* and a *base model* (see Section 3.3). The active model is the *policy* to be optimized, while the base model is a PLM. A Kullback-Leibler (KL) constant between the outputs of the active model and the based model ensures that the generated terms do not deviate far from the based model [41].

Experimental results on three widely-used datasets, SCIFACT, NQ, and MS-MARCO, demonstrate that QSparse alleviates mismatch issues in sparse retrieval and outperforms the traditional

[1]https://anonymous.4open.science/r/QSparse/.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

query expansion methods. Compared with LLM-based methods like query2doc (GPT3) and Q2D (Flan-T5), QSparse (GPT2) shows competitive results on SCIFACT and NQ. Moreover, we study the impact of using RL for fine-tuning PLMs in Section 4.5.1. The results show that RL substantially enhances the performance of PLMs compared to models solely fine-tuned through supervised learning across all three datasets. Thus, we believe that RL in QSparse can be seen as a subsequent fine-tuning process, further enhancing the performance of the PLM and contributing to improved query expansion results.

Our contributions are: **(1)** We propose QSparse, a novel query expansion method, to mitigate mismatch problems. Experimental results demonstrate that QSparse significantly enhances sparse retrieval performance on the SCIFACT and NQ datasets. A modest improvement was also observed on the MS-MARCO passage. **(2)** When compared with PLMs that have undergone supervised fine-tuning (without RL), QSparse enhances the evaluation score by approximately 1% across all three datasets. These results suggest that incorporating the RL mechanism within QSparse can potentially improve the performance of PLMs.

2 RELATED WORK

Query expansion mitigates mismatch problems in sparse retrieval by expanding new terms to the original query. Traditional query expansion methods focus on using lexical knowledge [4, 28, 37], or Pseudo-Relevance Feedback (PRF) such as RM3 [2]. PRF-based models assume the top retrieval documents are relevant to the query. This mechanism causes PRF-based models may fail if the initial set of retrieved documents is not highly relevant.

Recently, PLMs have been applied to query expansion, benefiting from their generated abilities and inherent knowledge [8, 21, 32, 40, 40]. For example, LLMs like GPT3, Flan-T5 and Flan-UL2 have been used in *query2doc* for query expansion [9, 38]. However, PLMs occasionally fail to manage the relevance and usefulness of their inherent information. Previous work uses RL to fine-tune PLMs on question-answering or conversation tasks[6, 7, 10, 39]. In this paper, we explore the potential of using PLM and RL for query expansion to address the mismatch between queries and documents in sparse retrieval methods. Also, we use Proximal Policy Optimization (PPO) [33] as our RL optimization algorithm, distinguishing it from previous studies [6, 7, 10, 39].

3 METHOD

3.1 PLM as Query Term Generator

We begin with the original query Q as $[q_1, q_2, \cdots, q_n]$, where q_i is the i-th term in the query. Let $T = [t_1, t_2, ..., t_m]$ be the list of query terms generated by a PLM. Then we create an expanded query by infixing a special separator token <SEP> as follows: $\hat{Q} = [q_1, q_2, ..., q_n, \text{<SEP>}, t_1, t_2, ..., t_m]$.

The probability of the generated token, $p(t_i)$, at time step i is defined as follows:

$$p(t_i) = \text{PLM}(\hat{Q}_{i-1}), \tag{1}$$

where $\hat{Q}_{i-1} = [q_1, q_2, ..., q_n, \text{<SEP>}, t_1, \cdots t_{i-1}]$.

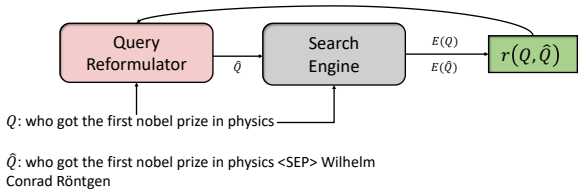

Figure 2: Reward function in QSparse.

3.2 Reward Function

We train the PLM under the paradigm of reinforcement learning human feedback (RLHF), except that we use evaluation metrics as the reward [36]. In RLHF, a reward function assigns a scalar value to pairs (query and response). This reward function can incorporate an evaluation metric, human feedback, or a combination of them. The underlying goal is to develop a function of receiving a text sequence and producing a scalar reward that accurately reflects human preferences [13]. We use the evaluation score of the information retrieval system, such as NDCG@10, as the reward function. Specifically, given the original query Q and expanded query $\hat{Q}$, we define our reward function r as:

$$r(Q, \hat{Q}) = E(\hat{Q}) - E(Q), \tag{2}$$

where $r(Q, \hat{Q}) \in \mathbb{R}$ and $E(\cdot)$ is the evaluation metric for the query. Eq. 2 shows that the reward function assigns a positive reward when the expanded query outperforms the original query. Figure 2 illustrates how the evaluation score is incorporated into our reward function and QSparse.

3.3 Fine-tuning PLMs with RL

QSparse uses the Proximal Policy Optimization (PPO) algorithm [33]. Given an original query Q from the dataset, T is generated by the current iteration of the fine-tuned policy, which is used to obtain the expanded query $\hat{Q}$, and a scalar score $r(Q, \hat{Q})$, as described in Section 3.1. The modified reward $R(Q, \hat{Q})$ sent to the RL update rule is defined as follows [41]:

$$R(Q, \hat{Q}) = r(Q, \hat{Q}) - \lambda \log \frac{\pi_{\text{PPO}}(T|Q)}{\pi_{\text{Base}}(T|Q)}, \tag{3}$$

where $\lambda \in \mathbb{R}^+$ is a hyperparameter that can either be constant or scheduled (i.e., decays with progressive steps). $\pi_{\text{PPO}}(T|Q)$ is the policy to be optimized by PPO while $\pi_{\text{Base}}(T|Q)$ is the based model. Note that the update rule is the parameter update from PPO that maximizes the reward metrics in the current batch of data (PPO is on-policy, which means the parameters are only updated with the current batch of prompt-generation pairs). PPO uses constraints on the gradient to ensure the update step does not destabilize the learning process [13].

4 EXPERIMENTAL RESULTS

4.1 Datasets

We have conducted experiments on three datasets: SCIFACT [35], NQ [12] and MS-MARCO passage retrieval [24], as depicted in Table 1. **SCIFACT** is a standard information retrieval test collection in Benchmarking-BEIR [35], which is a robust and heterogeneous

evaluation benchmark for information retrieval. We use the dataset format Beir[2]. **NQ** is an open-domain question answering dataset. Open-domain question answering relies on passage retrieval to select candidate contexts. We use the same dataset as in GAR [21]. **MS-MARCO** passage ranking task is a large-scale dataset focused on machine reading comprehension [24]. In the supervised fine-tuning process, we use the datasets released in the paper [38].

Table 1: Overview of the datasets used in our experiments.

Datasets	Train Pairs	Val queries	Test queries
SCIFACT	735	174	300
NQ-answer	79,168	8,757	3,610
NQ-title	69,790	7,514	3,610
NQ-sentence	79,168	8,757	3,610
MS-MARCO	502,939	6,980	6,838

4.2 Baselines and Comparison models

We use BM25 [31] as the sparse retrieval method. We compare QSparse with three strong baselines in each category of query expansion techniques: **(1)** a traditional PRF method (i.e., RM3 [1]), **(2)** LLM-based methods (i.e., Query2doc [38], Q2D [9]) and **(3)** LSTMs with RL (i.e., LSTM-RL [30].) Note that we implement all the models, except for the result of RM3 on NQ (from [21]) and on MS-MARCO passage (from [38]). All results of Query2doc are from [38]. We refer to Section 4.4 for implementation details.

The subscripts "I" and "II" used in Table 2 indicate the two types of reward scores utilized in the RL fine-tuning process. Note that the reward function uses one evaluation metric in each experiment. For instance, on the SCIFACT dataset, $\text{QSparse}_{\text{I}}$ and $\text{LSTM-RL}_{\text{I}}$ use NDCG@10 as the evaluation metric, while $\text{QSparse}_{\text{II}}$ and $\text{LSTM-RL}_{\text{II}}$ use MAP.

4.3 Evaluation Metrics

We use several metrics to evaluate the effectiveness of QSparse. For the SCIFACT dataset, we use the Normalised Discount Cumulative Gain (NDCG@10), which is the official evaluation in the BEIR [35]. We also use the Mean Average Precision (MAP) since there is no graded relevance judgment in the SCIFACT dataset. For the NQ dataset, we use the same recall metrics R@20 and R@100 used in [11]. For the MS-MARCO passage dataset, we employ the standard Mean Reciprocal Rank (MRR@10) and R@50 measures as used in [38].

4.4 Training Environment and Hyperparameters

Implementation. We implement QSparse using Transformer Reinforcement Learning (TRL) techniques, as described by [36]. We utilize the Pyterrier library implementation for BM25 and RM3 [20]. For Q2D, we use the released Flan-UL2 model used in the paper [9]. For LSTM-RL, we use the implementation provided in [27], with modifications to the reward function. For the DPR, we train a bi-encoder using the script released in BEIR [35]. The queries and documents are passed independently to the transformer network to produce fixed-sized embeddings. Evaluation in the training process is performed using the Pyterrier library [20], which allows for convenient query-specific evaluation scores during training. We conduct the training process on a single NVIDIA A-100 GPU machine.

Training process. Two steps are included: **(1)** In supervised fine-tuning, we fine-tune GPT2 based on a sequence-to-sequence paradigm. In the fusion analysis introduced in Section 4.5.2, we use the same fusion strategy as in [21]. Specifically, given a query, suppose three distinct models have retrieved three ordered lists of documents: $[a_1, a_2, ...]$, $[b_1, b_2, ...]$, and $[c_1, c_2, ...]$, respectively. The fusion list for this query is constructed by interleaving the documents from each model: $[a_1, b_1, c_1, a_2, b_2, c_2, ...]$. The supervised fine-tuning process involves 100 epochs, followed by selecting the best validation model. **(2)** In the RL fine-tuning process, we employed the model obtained through supervised fine-tuning as our initialized model. During training, we use RL to select expanded terms in original queries with the highest reward score.

Hyperparameters. For the SCIFACT dataset, we use the following hyperparameters during training: a batch size of 64, an input length of 20, and a target length of 20. For the NQ dataset, the input length is 20, and the target length is 40. In the reinforcement learning process, we have adopted the same training parameters used in [36].

4.5 Experimental Results

The experimental results are presented in Tables 2. **(1)** For SCIFACT and NQ datasets, QSparse significantly enhances the performance of BM25, as evidenced by improvements of 0.02 on NDCG@10 and 0.04 on MAP in SCIFACT, and 0.07 on R@20 and 0.04 on R@100 in NQ. Moreover, QSparse outperforms LSTM-RL, showing the advantage of RL combined with a PLM (instead of a small-scaled model like LSTM). Furthermore, QSparse outperforms LLMs-based methods (i.e., Query2doc and Q2D), underscoring the usefulness of RL in query expansion.**(2)** For MS-MARCO passage, our approach's performance is lower than that of Query2doc and Q2D. We speculate that this discrepancy arises from MS-MARCO being a more general topic dataset, which demands a greater amount of additional information stored in the model. Query2doc and Q2D employ LLMs (Flan-UL2 and GPT3, respectively) with larger parameter sizes and higher generation capabilities than GPT2 used in QSparse.

4.5.1 RL improvement. We analyze the improvements achieved through RL fine-tuning in Table 2 (see bottom). Compared to QSparse/PLM, which is solely fine-tuned through a supervised process, QSparse fine-tuned by RL (namely, $\text{QSparse-RL}_{\text{I}}$ and $\text{QSparse-RL}_{\text{II}}$) demonstrates greater efficacy across all three datasets. Therefore, RL fine-tuning can be considered a subsequent tuning process following the supervised fine-tuning of PLMs.

4.5.2 Dense Retrieval improvement. The combination of sparse retrieval and dense retrieval has shown usefulness in improving model effectiveness [18]. In this paper, we use a simple fusion strategy to combine Qparse with a classic dense retrieval method called DPR [11] (See Section 4.4 for an explanation of the fusion strategy).

[2]https://github.com/beir-cellar/beir

Table 2: Evaluation results on sparse retrieval methods using query expansion techniques. Superscripts † denote statistically significant ($p < .05$) improvements w.r.t BM25 in a standard t-test. For an explanation of the terminology used in the "Model" column, please refer to Section 4.2.

Model	Fine-tuning	SCIFACT		Neural Questions		MS-MARCO passage	
		NDCG@10	MAP	R@20	R@100	MRR@10	R@50
BM25	✗	0.684	0.638	0.629	0.781	0.184	0.604
+RM3[1]	✗	0.645	0.590	0.642	0.796	0.158	0.567
+Query2doc[38]	✗	0.653	-	-	-	**0.214**	**0.653**
+Q2D[9]	✗	0.681	0.635	0.672	0.803	0.195	0.626
+LSTM-RL$_{\text{I}}$[25]	✓	0.673	0.637	0.620	0.756	0.168	0.547
+LSTM-RL$_{\text{II}}$[25]	✓	0.691	0.651	0.619	0.755	0.168	0.547
+QSparse/PLM	✓	0.696	0.656	0.698†	0.821†	0.184	0.600
+QSparse$_{\text{I}}$	✓	**0.704**	**0.671**†	0.703†	**0.824**†	0.191	0.616
+QSparse$_{\text{II}}$	✓	0.701	0.664	**0.712**†	0.823†	0.191	0.615

Table 3: Evaluation results on dense retrieval methods. Superscripts ‡ denote statistically significant ($p < .05$) improvements w.r.t DPR in a standard t-test.

Model	SCIFACT		Neural Questions	
	NDCG@10	MAP	R@20	R@100
DPR[11]	0.728	0.706	0.795	0.861
Fusion	0.786‡	0.744‡	0.816‡	0.883‡

As shown in Table 3, we compare the performance of our fusion approach with that of DPR, on SCIFACT and NQ datasets. The results show that the fusion strategy significantly improves DPR's performance. Thus, we argue that although dense retrieval has shown excellent performance, the study of sparse retrieval is necessary because combining both can yield better results. Moreover, sparse retrieval holds significant advantages in terms of efficiency and interpretability.

4.6 Mean Response Time Analysis

To compare the efficiency of the query expansion techniques used in our experiments, we evaluate their *mean response time* on the NQ dataset. Running time is measured by the Pyserini library [16]. To ensure a fair comparison, all encoding processes were conducted with a single thread and a batch size of one (i.e., thread=1, batch-size=1). Figure 3 shows that QSparse outperforms all other query expansion techniques while maintaining a relatively high-efficiency level. QSparse strikes a balance between effectiveness and efficiency, making it well-suited for scenarios where there is a need for both efficiency and effectiveness.

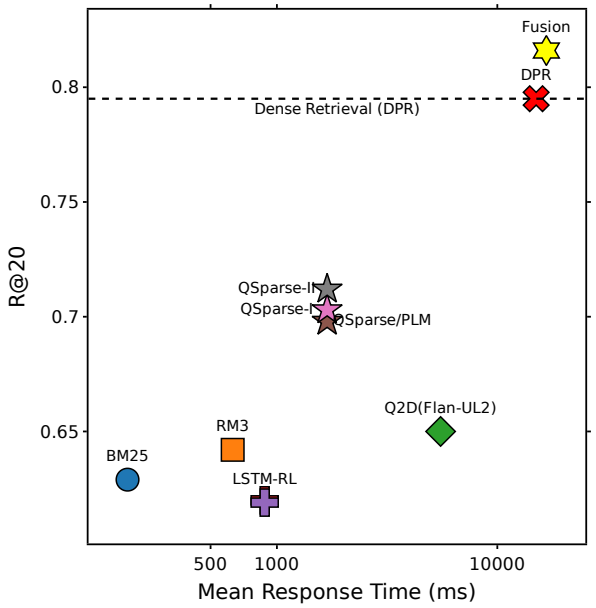

Figure 3: Mean response time evaluated on the NQ dataset. This response time is calculated by summing the time taken to generate terms and the subsequent search time.

5 CONCLUSION

We introduce a query expansion method called QSparse to address query-document term mismatch in sparse retrieval. Our experimental results demonstrate that QSparse significantly improves the performance of sparse retrieval. We argue that, in query expansion, RL fine-tuning can be considered a subsequent tuning process following the supervised fine-tuning of PLMs. This argument encourages further exploration and research in optimizing PLMs through RL-based approaches. The fusion analysis reveals that combining QSparse and dense retrieval leads to a considerable performance boost compared to using dense retrieval alone. Our findings indicate that while dense retrieval is generally more effective than sparse retrieval, there is still room for improvement by combining the two techniques. This area shall be explored in future work.

REFERENCES

[1] Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. *Computer Science Department Faculty Publication Series* (2004), 189.

[2] Giambattista Amati. 2003. *Probability models for information retrieval based on divergence from randomness.* PhD. University of Glasgow. https://eleanor.lib.gla.ac.uk/record=b2151999

[3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2204.05862* (2022).

[4] Jagdev Bhogal, Andrew MacFarlane, and Peter Smith. 2007. A review of ontology based query expansion. *Information processing & management* 43, 4 (2007), 866–886.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *arXiv preprint arXiv:1705.07830* (2017).

[7] Zhiyu Chen, Jie Zhao, Anjie Fang, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2022. Reinforced question rewriting for conversational question answering. *arXiv preprint arXiv:2210.15777* (2022).

[8] Ayyoob Imani, Amir Vakili, Ali Montazer, and Azadeh Shakery. 2019. Deep neural networks for query expansion using word embeddings. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part II 41*. Springer, 203–210.

[9] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query Expansion by Prompting Large Language Models. *arXiv preprint arXiv:2305.03653* (2023).

[10] Magdalena Kaiser, Rishiraj Saha Roy, and Gerhard Weikum. 2021. Reinforcement learning from reformulations in conversational question answering over knowledge graphs. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 459–469.

[11] 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).

[12] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.

[13] Nathan Lambert. 2022. *Illustrating Reinforcement Learning from Human Feedback (RLHF).* https://huggingface.co/blog/rlhf

[14] Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, Vol. 51. ACM New York, NY, USA, 260–267.

[15] Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, and Jungang Xu. 2018. NPRF: A neural pseudo relevance feedback framework for ad-hoc information retrieval. *arXiv preprint arXiv:1810.12936* (2018).

[16] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*. 2356–2362.

[17] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *arXiv preprint arXiv:2004.01909* (2020).

[18] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345.

[19] Yuanhua Lv and ChengXiang Zhai. 2009. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 255–264.

[20] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation inInformation Retrieval using PyTerrier. In *Proceedings of ICTIR 2020*.

[21] 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)*. 4089–4100.

[22] Donald Metzler and W Bruce Croft. 2007. Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 311–318.

[23] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).

[24] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *choice* 2640 (2016), 660.

[25] Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-Oriented Query Reformulation with Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 574–583. https://doi.org/10.18653/v1/D17-1061

[26] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155* (2022).

[27] Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* (2017).

[28] Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. 160–169.

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[30] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7008–7024.

[31] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[32] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608* (2016).

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[34] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems* 33 (2020), 3008–3021.

[35] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).

[36] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, and Nathan Lambert. 2020. TRL: Transformer Reinforcement Learning. https://github.com/lvwerra/trl.

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

[38] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. *arXiv preprint arXiv:2303.07678* (2023).

[39] Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2021. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. *arXiv preprint arXiv:2112.08558* (2021).

[40] Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: contextualized query expansion for document re-ranking. *arXiv preprint arXiv:2009.07258* (2020).

[41] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593* (2019).