QRPLM: Query Reformulation using Pre-trained Language Model with Reinforcement Learning

Anonymous Author(s)*

ABSTRACT

In spite of substantial efforts, the issue of mismatching between query and document in sparse retrieval persists. To address this problem, we propose a novel query formulation named QRPLM based on Reinforcement Learning from Human Feedback (RLHF). Specifically, the pre-trained language model is employed as a query reformulator under the framework of reinforcement learning, where the reward is based on the evaluation metric. Our evaluation of QRPLM on two widely-used information retrieval datasets, namely SCIFACT and Neural Question (NQ), demonstrates that QRPLM can enrich the original query and significantly outperform the sparse retrieval. Furthermore, the fusion of QRPLM and dense retrieval results in an improvement of 8% on SCIFACT in terms of NDCG@10 and 2% on NQ in terms of recall, as compared to the original dense retrieval. Notably, our evaluation results demonstrate that QRPLM enables the utilization of the advantages of both sparse retrieval and dense retrieval in addressing the mismatch issue. Our solution and code are publicly available on GitHub[1].

CCS CONCEPTS

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

KEYWORDS

query reformulation, pre-trained language model, information retrieval

ACM Reference Format:

Anonymous Author(s). 2018. QRPLM: Query Reformulation using Pre-trained Language Model with Reinforcement Learning. 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 retrievals involves matching the words in a given query with those in relevant documents. Despite the effectiveness of this approach, it is prone to the problem of mismatch, which can result in the omission of relevant documents [18]. As a consequence, the efficacy of sparse retrieval can be compromised by the presence of mismatch issues.

[1]https://anonymous.4open.science/r/trl-C1A1/.

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

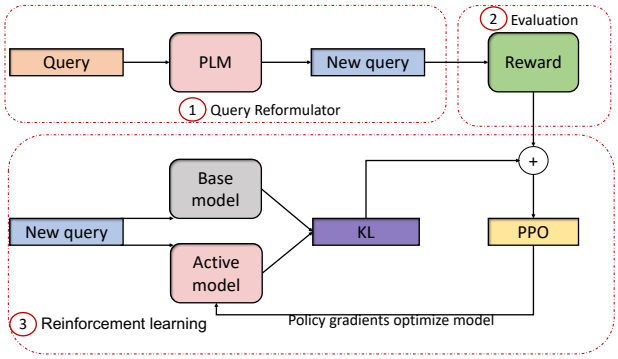

Figure 1: QRPLM in RHLF overview.

One way to address this problem is to conduct *automatic query reformulation*, which aims to enhance the retrieval effectiveness of the query by adding more relevant words. The technique typically select relevant words from WordNet [16], or retrieval documents known as the *Pseudo Relevance Feedback* (PRF) [8, 9, 11, 15, 29].

Rodrigo et al. first frame the query reformulation as a Reinforcement Learning (RL) problem and estimates the upper-bound performance of an RL-based model, showing its potential for future improvements [18]. Inspired by their work, we introduce a novel query reformulation approach named QRPLM, using Reinforcement Learning from Human Feedback (RLHF) paradigm. Recently, the RLHF paradigm has shown great success in natural language processing [3, 17, 19, 25, 30]. The much-vaunted dialogue tool ChatGPT applies RLHF on dialogue task [7], showcasing its great potential.

Figure 1 shows an overview of the RHLF paradigm for QRPLM, which comprises three distinct steps. **(1)**A query reformulator is utilized, wherein a pre-trained language model (PLM) [21] generates terms based on the original query. The original query augmented with generated terms is considered the new query, and this process is referred to as query reformulation (see Section 3.1). **(2)** Evaluation: The query and generated terms are evaluated with a reward function to yield a scalar value. We involve the evaluation metric for each query in the reward function (see Section 3.2). **(3)** Reinforcement learning optimization: During training, pairs (query, generated terms) are used to calculate the log probabilities of the terms in their sequences. In this approach, 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 pre-trained language model. 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 language model [30]. Note that the optimization algorithm is Proximal Policy Optimization (PPO) [24].

Our experimental evaluation on two distinct information retrieval datasets, SCIFACT and Natural Question, has shown that our proposed method, QRPLM, can effectively augment the original

query and alleviate the mismatch problem. Our evaluation results have demonstrated that QRPLM significantly improves sparse retrieval and outperforms other query reformulation methods. In addition to this, we have combined the results of QRPLM with dense retrieval. Our fusion results have achieved the best performance in comparison to the other models, which suggests that QRPLM is complementary to both sparse retrieval and dense retrieval and can benefit both.

Our **contributions** are three folds: 1) We propose a novel query reformulation approach (QRPLM) based on RLHF. 2) The experimental results show that QRPLM can enrich the original query and mitigate the mismatch problems in sparse retrieval. The evaluation results on the two datasets show that QRPLM can significantly improve sparse retrieval and outperform other query reformulation methods. 3) The fusion results with QRPLM and dense retrievel achieve the best results, demonstrating that QRPLM enables the utilization of the advantages of both sparse retrieval and dense retrieval in addressing the mismatch issue.

2 RELATED WORK

Our work is broadly related to two general areas:

Reinforcement learning from Human Feedback. Zieglar et al. take the first step to apply RLHF to four natural language tasks: continuing text with positive sentiment or physically descriptive language, and summarization tasks on the TL;DR and CNN/Daily Mail datasets [30]. Nisan et al. apply RLHF to summarizing text that significantly outperforms human reference summaries and much larger models fine-tuned with supervised learning alone [25]. Reiichiro et al. introduce a WebGPT using RLHF to fine-tune GPT-3 to answer long-form questions using a text-based web browsing environment [17]. Long et al. apply RLHF to a general language model [19]. Jacob et al. train a language model with RLHF to return answers with specific citations [14]. Yuntao et al. propose detailed documentation of training a language model assistant with RLHF [3]. Deborah et al. use RL to enhance the conversational skill of an open-ended dialogue agent [4]. Open AI proposes a ChatGPT under the framework of RLHF for suitable use as an all-purpose chat bot [7].

Query Reformulation. This approach is typically based on query expansion and query rewrite. Traditional query expansion methods, including RM3 [1] expand query based on the likelihood that terms in the top-ranked documents co-occur with the original query term [2]. Another direction is to avoid PRF and instead reformulate queries directly with the encoder-decoder paradigm. For instance, with the aid of separate fields in documents, Mao et al. train Generation-Augmented Retrieval (GAR) that generates the title of a relevant text given a query and appends the generated text to the query [13]. Not only enriching queries semantics, Seq2Seq models are used to reduce noise in queries. This can be achieved by simplifying natural language queries into keyword queries [10] or even generating paraphrases of queries [28]. Nogueira et al. first apply reinforcement learning to query formulation tasks. They use evaluation system scores for information retrieval to get the reward scores [18].

Compared with previous studies, our model QRPLM is the first work to apply RLHF to query reformulation tasks. We use the the pre-trained language model as a query reformulator and fine-tuned the model using reinforcement learning.

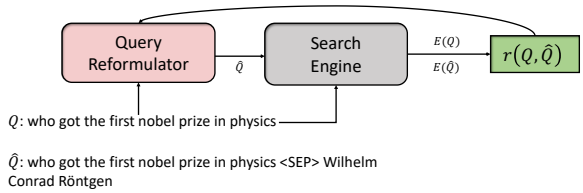

Figure 2: Reward function in QRPLM.

3 METHOD

The QRPLM method, depicted in Figure 1, is composed of three primary steps: **(1)** utilizing the PLM as a query reformulator, **(2)** employing a reward function, and **(3)** fine-tuning the PLM through reinforcement learning (RL). The following subsections provide detailed explanations of each of these individual steps.

3.1 PLM as Query reformulator

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 pre-trained language model (PLM) as the *query reformulator*. Then, we create a reformulated 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 $\hat{Q}_{i-1} = [q_1, q_2, ..., q_n, \text{<SEP>}, t_1, \cdots t_{i-1}]$.

3.2 Reward Function

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 [27]. The underlying goal is to develop a function of receiving a text sequence and producing a scalar reward that accurately reflects human preferences [7].

QRPLM employs an evaluation metric, such as NDCG@10, as the reward function. Specifically, given the original query Q and reformulated 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 a positive reward is assigned by the reward function when the reformulated query outperforms the original query. Figure 2 illustrates how the evaluation score be incorporated into our reward function and the QRPLM. The advantage of using evaluation metrics instead of human feedback as a reward is that the differing values of humans cause these scores to be uncalibrated and noisy [7].

3.3 Fine-tuning PLM with RL

The policy optimization technique adopted in QRPLM is the Proximal Policy Optimization (PPO) algorithm [24].

Given a original query Q from the dataset, T is generated by the current iteration of the fine-tuned policy, which is used to obtain the reformulated query $\hat{Q}$, and a scalar score $r(Q, \hat{Q})$, as described in Sec. 3.1. The modified reward $R(Q, \hat{Q})$ sent to the RL update rule is defined as follows [30]:

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

where $\lambda \in \mathbb{R}^+$ is a hyperparameter that can either be constant or scheduled (decay with step). $\pi_{\text{PPO}}(T|Q)$ 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 [7].

4 EXPERIMENTAL RESULTS

4.1 Datasets

We have conducted experiments on two datasets: SCIFACT [26] and Natural Question [6], as depicted in Table 1.

SCIFACT is a standard information retrieval test collection in Benchmarking-BEIR [26], which is a robust and heterogeneous evaluation benchmark for information retrieval. We use the dataset format released in [2]. The dataset consists of 919 query target pairs, which are divided into two parts, with 80% utilized as the training set and 20% for the validation set. The test set is the same as the dataset used in BEIR [26].

Natural Question (NQ) is an open-domain question answering (OpenQA) dataset. Open-domain question answering relies on passage retrieval to select candidate contexts. We used the same dataset as in GAR [13]. There are three types of contexts that can be used as a ground truth target.

Table 1: Datasets 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

4.2 Baselines and Comparision models

We compare QRPLM with classic sparse retrieval (BM25 [23]), dense retrieval (DPR [5]) and other traditional query reformulation methods (RM3, LSTM-RL [22]).

- **BM25** is a classic sparse retrieval based on a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document [23].

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

Table 2: Statistics of SCIFACT and NQ.

	Mean	Min	50%	75%	90%	Max
SCIFACT						
Query Length	12	3	12	15	19	39
Document Length	220	58	174	275	369	692
Neural Question						
Query Length	9	8	9	10	11	23
NQ-answer	3	1	2	4	5	83
NQ-title	25	1	26	40	49	140
NQ-sentence	38	1	28	42	72	351

- **DPR** encodes queries and documents to embeddings. Then the model learned from a simple dual encoder framework [5].
- **RM3** is a traditional PRF method [1], which expands queries based on the likelihood that terms are in the top-ranked documents.
- **LSTM-RL**$_{(\cdot)}$ is a model using LSTM as a query reformulator and training with self-critical policy gradient training algorithm for RL [22], where $(\cdot)$ denotes different reward functions. This algorithm is the same as [20].
- **Fusion-QRPLM** is the fusion of the results from QRPLM and dense retrieval. We adopt the fusion strategy employed in a prior work [13]. Specifically, given two ranked document sets obtained from QRPLM and DPR, the fused set selects one top-ranked document from each of the two sets alternately.

4.3 Evaluation Metrics

We use several metrics to evaluate the *effectiveness* of the QRPLM. For the SCIFACT dataset, we use the Normalised Discount Cumulative Gain (NDCG@10), which is the official evaluation in the BEIR [26]. 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 Recall@20 and Recall@100 used in [5].

4.4 Training Environment and Hyperparameters

Implementation. We implement QRPLM using Transformer Reinforcement Learning (TRL) techniques, as described by [27]. We utilized the Pyterrier library implementation for BM25 and RM3 [12]. For LSTM-RL, we used the implementation provided in [20], with modifications to the reward function. For the DPR, we trained a bi-encoder using the script released in BEIR [26], the queries and documents are passed independently to the transformer network to produce fixed-sized embeddings. The evaluation in the training process was performed using the Pyterrier library [12], which allowed for convenient query-specific evaluation scores during training. The training process was conducted on a single NVIDIA A-100 GPU machine.

Training process. Two steps are included: **(1)** In supervised fine-tuning, we initially fine-tuned a pre-trained language model GPT using SCIFACT and NQ. This sequence-to-sequence paradigm involved three ground truths for NQ, namely answer, title, and

sentence, which required using three separate models. We use the same fusion strategy in [13]. For each query, suppose we have three document lists $[a_1, a_2, ...]$, $[b_1, b_2, ...]$, $[c_1, c_2, ...]$ have been recalled. Then the fusion list for this query is $[a_1, b_1, c_1, a_2, b_2, c_2, ...]$. The supervised fine-tuning process involved 100 epochs, followed by the selection of the best validation model. The resulting based model is introduced in Section 3.3. **(2)** In the RL fine-tuning process, we employed the model obtained through supervised fine-tuning as our initialized model. During training, only the query in the training dataset was used, and the best model with the highest reward score was selected. We fused the results from three contexts in NQ [13].

Hyperparameters. For the SCIFACT dataset, the hyperparameters used during training are as follows: 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 as used in [27].

4.5 Experimental Results

Table 3: Evaluation on SCIFACT. Superscripts † and ‡ denote statistically significant ($p < .05$) improvements w.r.t BM25 and DPR in a standard t-test.

QR model	Ranking Model	NDCG@10	MAP
Baselines			
None	Sparse(BM25)	0.684	0.638
None	Dense(DPR)	0.728	0.706
RM3	Sparse(BM25)	0.645	0.590
LSTM-RL$_{\text{NDCG}}$	Sparse(BM25)	0.673	0.637
LSTM-RL$_{\text{MAP}}$	Sparse(BM25)	0.691	0.651
QRPLM$_{\text{NDCG}}$	Sparse(BM25)	0.704	0.671†
QRPLM$_{\text{MAP}}$	Sparse(BM25)	0.701	0.664
Fusion-QRPLM	BM25+DPR	**0.786**†‡	**0.744**†‡

Table 4: Evaluation on Natural Question. Superscripts † and ‡ denote statistically significant ($p < .05$) improvements w.r.t BM25 and DPR in a standard t-test.

QR model	Ranking Model	R@20	R@100
Baselines			
None	Sparse(BM25)	0.629	0.781
None	Dense(DPR)	0.795	0.861
RM3	Sparse(BM25)	0.642	0.796
LSTM-RL$_{\text{R@20}}$	Sparse(BM25)	0.620	0.756
LSTM-RL$_{\text{R@100}}$	Sparse(BM25)	0.619	0.755
QRPLM$_{\text{R@20}}$	Sparse(BM25)	0.703†	0.824†
QRPLM$_{\text{R@100}}$	Sparse(BM25)	0.712†	0.823†
Fusion-QRPLM	BM25+DPR	**0.816**†‡	**0.883**†‡

The experimental results on SCIFACT and NQ are presented in Tables 3 and 4, respectively. Our proposed query reformulation model QRPLM 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, QRPLM outperforms alternative query reformulation methods, such as RM3 and LSTM-RL. Across both SCIFACT and NQ experiments, the dense retrieval model DPR exhibits substantially better performance than the traditional sparse retrieval model (BM25). Furthermore, we observe that fusing the results of QRPLM with dense retrieval leads to a significant performance improvement, as demonstrated by the Fusion-QRPLM result. This finding suggests that the fusion of sparse and dense retrievals under the QRPLM framework enables the utilization of their respective advantages in addressing the mismatch issue.

4.6 Case Study

To further investigate how QRPLM addresses the mismatch issue in sparse retrieval, we conducted a case study. As demonstrated in Table 5, there is no overlap between the query and the relevant document, which renders it impossible for sparse retrieval to retrieve the relevant document. However, QRPLM generates relevant terms that address this problem by adding them to the query.

Table 5: Example of terms generated comparison. Blue context is the exact match term between the query and the relevant document.

Query: 0-dimensional biomaterials show inductive properties **Generated terms by LSTM**: [unk] outer membrane **Generated terms by QRPLM**: nanotechnologies emerging platforms could **Relevant document**: New opportunities: the use of nanotechnologies to manipulate and track stem cells. Nanotechnologies are emerging platforms that could be useful ...

5 CONCLUSION

We introduce a query reformulation method called QRPLM that leverages the RLHF paradigm to address the mismatch issue in sparse retrieval. Our experimental results demonstrate that QRPLM significantly improves the performance of sparse retrieval and performs better than alternative query reformulation approaches. Additionally, our fusion analysis reveals that the combination of QRPLM and dense retrieval leads to a considerable performance boost compared to using dense retrieval alone. These findings demonstrate that the fusion of sparse and dense retrievals under the QRPLM framework allows for utilizing their respective advantages in addressing the mismatch issue. Furthermore, this indicates 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] Deborah Cohen, Moonkyung Ryu, Yinlam Chow, Orgad Keller, Ido Greenberg, Avinatan Hassidim, Michael Fink, Yossi Matias, Idan Szpektor, Craig Boutilier, et al. 2022. Dynamic Planning in Open-Ended Dialogue using Reinforcement Learning. *arXiv preprint arXiv:2208.02294* (2022).

[5] 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).

[6] 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.

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

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

[9] 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).

[10] Xiaoyu Liu, Shunda Pan, Qi Zhang, Yu-Gang Jiang, and Xuanjing Huang. 2018. Generating Keyword Queries for Natural Language Queries to Alleviate Lexical Chasm Problem. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1163–1172. https://doi.org/10.1145/3269206.3271727

[11] 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.

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

[13] Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553* (2020).

[14] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147* (2022).

[15] 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.

[16] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[17] 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).

[18] 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

[19] 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).

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

[21] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[22] 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.

[23] 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.

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

[25] 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.

[26] 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).

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

[28] Qian Yang, Zhouyuan Huo, Dinghan Shen, Yong Cheng, Wenlin Wang, Guoyin Wang, and Lawrence Carin. 2019. An End-to-End Generative Architecture for Paraphrase Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3132–3142. https://doi.org/10.18653/v1/D19-1309

[29] 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).

[30] 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).