# Discrete Prompt Optimization via Constrained Generation for Zero-shot Re-ranker

**Sukmin Cho** **Soyeong Jeong** **Jeongyeon Seo** **Jong C. Park***
School of Computing
Korea Advanced Institute of Science and Technology
{nelllpic,starsuzi,yena.seo,jongpark}@kaist.ac.kr

## Abstract

Re-rankers, which order retrieved documents with respect to the relevance score on the given query, have gained attention for the information retrieval (IR) task. Rather than fine-tuning the pre-trained language model (PLM), the large-scale language model (LLM) is utilized as a zero-shot re-ranker with excellent results. While LLM is highly dependent on the prompts, the impact and the optimization of the prompts for the zero-shot re-ranker are not explored yet. Along with highlighting the impact of optimization on the zero-shot re-ranker, we propose a novel discrete prompt optimization method, **Co**nstrained **Prompt** generation (Co-Prompt), with the metric estimating the optimum for re-ranking. Co-Prompt guides the generated texts from PLM toward optimal prompts based on the metric without parameter update. The experimental results demonstrate that Co-Prompt leads to outstanding re-ranking performance against the baselines. Also, Co-Prompt generates more interpretable prompts for humans against other prompt optimization methods.

## 1 Introduction

Information retrieval (IR) is the task of searching for documents relevant to a given query from a large corpus. As re-ranking the fetched documents from the retriever effectively enhances the performance and the latency, recent studies have suggested several kinds of re-rankers by fine-tuning pre-trained language models (PLM) (Nogueira and Cho, 2019; Nogueira et al., 2020). Furthermore, Sachan et al. (2022) show that large-scale language models (LLMs) such as GPT-3 (Brown et al., 2020) can be exploited as a zero-shot re-ranker with the prompt describing the task. They also highlight the importance of an appropriate prompt to elicit the full performance of LLMs, rather than updating the parameters. They choose an optimal prompt among the handcrafted candidates by cross-validation. However, such a manual search for the discrete prompts is highly expensive and sub-optimal in transferability.

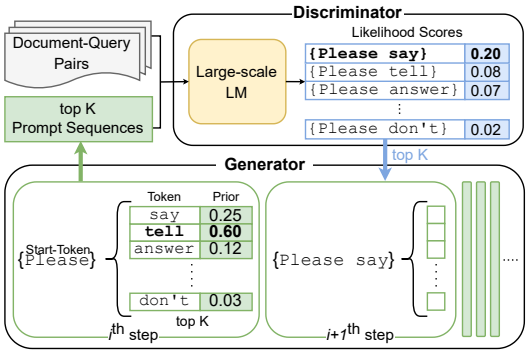


Figure 1: An overview of the constrained prompt generation process.

To resolve the issue, several methods are proposed for automatically optimizing the discrete prompt. They focus on text classification or mask-filling task while underestimating the open-ended generation (Shin et al., 2020; Gao et al., 2021; Prasad et al., 2022). Recently, Deng et al. (2022) address the discrete prompt optimization applicable to generation tasks with reinforcement learning by designing the reward function, which measures the generated text belonging to a discrete label. Since there are tasks that are still not aligned, requiring a continuous score of output, we aim at a prompt optimization for one of such tasks: re-ranking.

In this paper, we propose **Co**nstrained **Prompt** generation, Co-Prompt, as left-to-right discrete prompt optimization without additional model training. By defining the metric of prompt optimum for re-ranking, we interpret the searching process of the optimal prompt as constrained generation with two modules: a zero-shot re-ranker as a discriminator and any decoder-only PLM as a generator. The discriminator calculates the likelihood (i.e., metric) that the prompt sequence is optimal for guiding an LLM to distinguish relevant

* Corresponding author



*Findings of the Association for Computational Linguistics: ACL 2023*, pages 960–971
July 9-14, 2023 

documents among the large set for a given query. The generator samples the prompt tokens having a high prior from the previous prompt sequences for effectively restricting the prompt candidates for discriminator to evaluate. An overview of Co-Prompt is shown in Figure 1.

We validate our method, Co-Prompt, against other optimization baselines on two LLMs, T0 (Sanh et al., 2022) and OPT (Zhang et al., 2022), with two benchmark datasets, MS-MARCO (Nguyen et al., 2016) and Natural Question (Kwiatkowski et al., 2019). Experimental results show that Co-Prompt consistently generates well-performing prompts regardless of LLMs and datasets over the baselines. The qualitative analyses also support the interpretability of the prompts generated by Co-Prompt, similar to human language patterns.

Our contributions in this work are threefold:

- We highlight the impact of optimal prompt on a zero-shot re-ranker by exploiting the optimization methods.
- We propose Co-Prompt, a novel discrete prompt optimization via constrained generation for a zero-shot re-ranker.
- We experimentally show that Co-Prompt consistently guides the re-ranker well against the baselines and its output is similar to human language patterns.

## 2 Related Work

**Document Ranking with Generative Model** Using the generative model is one of the dominant methods for ranking the retrieved documents by defining the relevance score as the query likelihood score (Nogueira dos Santos et al., 2020; Ju et al., 2021). More recently, Sachan et al. (2022, 2023) showed that the LLM serves as either a zero-shot re-ranker or a training module of an unsupervised dense retriever. However, unlike ours, they require carefully designed manual prompts, which may have a limitation in transferability.

**Prompt Optimization** As prompting is considered a key variable when exploiting LLMs for various NLP tasks, finding the optimal prompt has become important to get the best performance out of the LLMs (Kojima et al., 2022; Xie et al., 2022). Recently, the prompt optimization work has focused on discrete prompt search (Shin et al., 2020; Gao et al., 2021; Deng et al., 2022) or soft prompt learning over a continuous space (Liu et al., 2021; Qin and Eisner, 2021; Lester et al., 2021). While the existing optimization methods mainly consider text classification or mask-filling task, their applicability to re-ranking is yet underexplored. In this paper, we target at optimizing discrete prompts for zero-shot re-ranker to get higher relevance scores for more relevant pairs via constrained generation.

**Constrained Generation** Constrained generation aims at deriving the text sequences that follow a certain constraint (Keskar et al., 2019). Utilizing a discriminator for guiding the generation toward the constraint via the Bayes' rule is one of the widely used constraint generation methods (Dathathri et al., 2020; Krause et al., 2021; Chaffin et al., 2022). Inspired by the effectiveness of the discriminator-based method, we adopt the zero-shot re-ranker as a discriminator when generating optimal discrete prompt sequences.

## 3 Method

### 3.1 Preliminaries

An LLM re-ranks the retrieved document $d$ concerning the relevance score with a given query $q$ as the query generation score:

$$\begin{aligned}\log P(d|q) &\propto \log P(q|d,\rho) \\ &= \frac{1}{|q|}\sum_t \log P(q_t|q_{<t},d,\rho),\end{aligned} \tag{1}$$

where $|q|$ denotes the token length of the query $q$ and $\rho$ is a natural language prompt guiding an LLM to generate the query $q$. Since the prompt $\rho$ is the only controllable variable in Equation 1, searching for an optimal prompt is a simple yet effective way to enhance the performance of LLMs. Thus, in this work, we focus on a prompt optimization strategy.

### 3.2 Constrained Prompt Generation

We define the optimal prompt $\rho^*$ for the re-ranker which maximizes the query generation scores:

$$\rho^* = \arg\max_{\rho} \mathbb{E}_{(d_i,q_i)\in D}[P(q_i|d_i,\rho)], \tag{2}$$

where $D$ is the dataset for the retriever, consisting of pairs of a query and its relevant document.

We solve the task of searching the optimal prompt $\rho^*$ for the document-query pair dataset $D$ with discriminator-based constrained generation. The generation is guided by the Bayes' rule:

$$P(\rho_t|D,\rho_{1:t-1}) \propto P_{M_D}(D_s|\rho_{1:t})P_{M_G}(\rho_t|\rho_{1:t-1}), \tag{3}$$

**Algorithm 1:** Co-Prompt: a beam search-based prompt generation algorithm with a discriminator and a generator. $D_s$: document-query pairs, $B$: beam width, $L$: maximum prompt length, $N$: the number of final prompts, $\mathcal{V}$: vocabulary set

**Require:** $D_s, B, L, \mathcal{V}$
**begin**
  $P_1 \leftarrow \{\text{Start-Token}\}$
  **for** $t = 1, \ldots, L$ **do**
    $P_{t+1} \leftarrow \emptyset$
    **foreach** $\rho_{1:t} \in P_t$ **do**
      $S_{t+1} \leftarrow \underset{K=B, \rho_{t+1} \in \mathcal{V}}{topK} P_{M_G}(\rho_{t+1}|\rho_{1:t})$
      $P_{t+1} \leftarrow P_{t+1} \cup \{\rho_{1:t+1} | \rho_{1:t} \oplus \rho_{t+1} \in S_{t+1}\}$
    **end**
    $P_{t+1} \leftarrow \underset{K=B, \rho_{1:t+1} \in P_{t+1}}{topK} P_{M_D}(D_s|\rho_{1:t+1})$
  **end**
  $P \leftarrow \cup_{t \in [1,L]} P_t$
  $R \leftarrow \underset{K=N, \rho \in P}{topK} P_{M_D}(D_s|\rho)$
  **return** $R$
**end**

where $M_D$ is a zero-shot re-ranker serving as a discriminator, $M_G$ is a decoder-only PLM as a generator, and $D_s$ is a dataset sampled from $D$.

**Discriminator** The discriminator $M_D$ measures how effectively the prompt sequence $\rho_{1:t}$ guides the zero-shot re-ranker to generate the query from the given document by computing the likelihood $P_{M_D}(D_s|\rho)$, defined as the expectation of relevance score between document-query pairs $(q_i, d_i)$ of the sampled dataset $D_s$ with the prompt $\rho$:

$$P_{M_D}(D_s|\rho) = \mathbb{E}_{(d_i,q_i)\in D_s}[P_{M_D}(q_i|d_i, \rho)]. \quad (4)$$

We use this likelihood as the metric for prompt optimum. The other option of $P_{M_D}$ is shown in Appendix B.1.

**Generator** The generator $M_G$ samples the pool of prompts to be evaluated by a discriminator since computing Equation 3 of all possible tokens in the vocabulary requires a prohibitively high computational cost. The decoder-only PLM is exploited to sample prompt tokens $\rho_t$ having a high prior $P_{M_G}(\rho_t|\rho_{1:t-1})$ in a zero-shot manner.

We combine these modules to optimize the prompt by iteratively performing two steps: candidate generation and evaluation. We choose to use a beam search as a decoding strategy for left-to-right prompt generation. The detailed steps of the decoding strategy are shown in Algorithm 1.

## 4 Experimental Setups

We describe the experimental setups for validating the performance of the prompts. Our code is publicly available at github.com/zomss/Co-Prompt.

**Datasets** We employ two information retrieval datasets: **1) MS-MARCO** (Nguyen et al., 2016), collected from the Bing search logs, and **2) Natural Question (NQ,** Kwiatkowski et al. (2019)), fetched from Google search engines. We only use the document data of the dataset for evaluation. More information is shown in Appendix A.1.

| | NQ | | | | MS-MARCO | | | |
|---|---|---|---|---|---|---|---|---|
| | BM25 | | DPR | | BM25 | | DPR | |
| ACC@$k$(→) | 20 | 100 | 20 | 100 | 20 | 100 | 20 | 100 |
| Only Retriever | 62.9 | 78.3 | 79.2 | 85.7 | 48.0 | 66.7 | 37.5 | 55.5 |
| *T0-3B Re-ranker* | | | | | | | | |
| Null Prompt | 73.1 | 82.8 | 78.5 | 86.6 | 53.2 | 72.7 | 51.5 | 68.0 |
| P-tuning | 72.8 | 82.7 | 79.1 | 87.0 | 54.1 | 72.5 | 52.5 | 68.2 |
| RL Prompt | 74.7 | 83.4 | 79.9 | 87.4 | 60.9 | 77.4 | 57.1 | 71.2 |
| Manual Prompt | **75.7** | **83.8** | **81.3** | **87.8** | 60.6 | 77.9 | 57.7 | **72.0** |
| Co-Prompt (Ours) | 75.0 | **83.8** | 80.4 | 87.7 | **61.9** | **78.0** | **58.0** | 71.7 |
| *OPT-2.7B Re-ranker* | | | | | | | | |
| Null Prompt | 70.5 | 81.9 | 76.3 | 86.1 | 50.4 | 71.7 | 50.1 | 68.1 |
| P-tuning | 71.2 | 82.8 | 78.3 | 87.5 | 56.5 | 75.5 | 54.6 | 69.9 |
| RL Prompt | 72.5 | 82.9 | 79.1 | 87.4 | 59.2 | 76.7 | 56.3 | 71.1 |
| Manual Prompt | 73.1 | 83.3 | 78.9 | 87.2 | 55.3 | 74.6 | 54.3 | 70.1 |
| Co-Prompt (Ours) | **75.2** | **84.1** | **80.2** | **88.1** | **59.3** | **77.2** | **56.4** | **71.3** |

Table 1: ACC@$k$ of the re-ranked result with the prompts when $k$ is 20 and 100. The best scores are marked in **bold**, and the next ones are underlined.

**Evaluation Metrics** We evaluate the results by two metrics, ACC and nDCG. **1) ACC** is the percentage of the relevant documents in the total retrieved ones. **2) nDCG**, normalized discounted cumulative gain, reflects that the more relevant documents should record higher ranks.

**Retriever & Re-ranker** We select two widely used sparse and dense retrievers as our retrievers, which are **1) BM25** (Robertson and Zaragoza, 2009) and **2) DPR** (Karpukhin et al., 2020), respectively. For the zero-shot re-ranker, we use **1) T0** (Sanh et al., 2022) and **2) OPT** (Zhang et al., 2022). We describe more detailed information in Appendix A.3 and A.4.

**Prompt Baselines** We compare Co-Prompt against four baselines: **1) Null Prompt** is an empty prompt without any token. **2) P-Tuning** is a soft prompt optimization method that yields prompt embeddings from the prompt encoder (Liu et al., 2021). **3) RL-Prompt** is a discrete prompt optimization method by training policy network (Deng et al., 2022). Note that we modify RL-Prompt and P-Tuning applicable to the re-ranking task. **4) Manual Prompt**, suggested by Sachan et al. (2022), is given as "*Please write a question based on this passage*", following the assumption that it is one of the best prompts that humans can find. Last, **5) Co-Prompt**, our proposed method, is a discrete prompt optimization method in left-to-right zero-shot generation. The implementation details of baselines are shown in Appendix A.5.

| Retriever \Re-ranker | Prompt | MS-MARCO Instruction Prompt | nDCG | NQ Instruction Prompt | nDCG |
|---|---|---|---|---|---|
| BM25 | - | - | 25.2 | - | 20.2 |
| OPT | Manual Prompt | *"Please write a question based on this passage"* | 28.7 | *"Please write a question based on this passage"* | 27.9 |
| | RL-Prompt | *"questions answers key question defining"* | 31.5 | *" poll trivia trivia wondered asking"* | 27.2 |
| | Co-Prompt | *"Please tell that• is the first question asked on Google for"* | **31.9** | *"Please post your question again when its not just about"* | 30.6 |
| | | *"Score! What are all 3 things, the first is"* | 30.2 | *"Score the top 5 things on this sub reddit for"* | 29.3 |
| | | *"This looks like the same as every "what are the"* | 30.5 | *"This post should be titled as"* | **31.2** |
| | | *"What are some common questions asked on the internet about"* | 30.3 | *"How do i find the name on google, and"* | 29.1 |

Table 2: Comparison of different discrete prompts and evaluation on the top-20 documents retrieved by BM25. The best results of each re-ranker are marked in **bold**.

| Retriever \Re-ranker | Prompt \Generator | MSMARCO nDCG@20 | nDCG@100 |
|---|---|---|---|
| BM25 | - | 22.84 | 28.70 |
| T0 | GPT2-Base | 30.76 | 36.44 |
| | GPT2-Large | **31.11** | **36.79** |
| | GPT2-XL | 29.86 | 35.71 |

Table 3: Comparison between the prompts from the different generators. The best results are marked in **bold**.

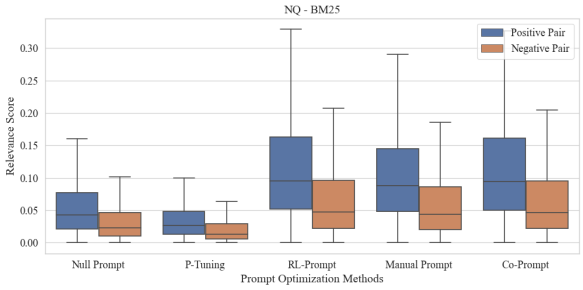


Figure 2: Distributions of relevance scores between document-query pairs. The positive pairs mean relevant ones and the negative pairs irrelevant.

**Implementation Details** The discriminator $M_D$ is the same model as the zero-shot re-ranker. Since the generator $M_G$ should be a decoder-only model, in the case of T0, GPT2-Large (Radford et al., 2019) is utilized as the generator. OPT, a decoder-only model, is used as both the discriminator and the generator. We use the start token as "Please" for a direct comparison with the manual prompt and fix the beam width $B$ as 10 and the maximum prompt length $L$ as 10 in our experiment.

**Environment** We conduct all experiments including prompt searching and document re-ranking on V100 32GB GPUs. We use BEIR (Thakur et al., 2021) framework[1] for re-ranked result evaluation and passage retrieval datasets. Also, the retrievers, BM25 and DPR, are from the same framework. We employ T0 and OPT with 3B and 2.7B parameters each for the discriminator and the re-ranker publicly open on the Huggingface model hub[2] (Wolf et al., 2020).

## 5 Result

In this section, we show the overall results of our method, Co-Prompt, with a detailed analysis.

**Overall Results** As shown in Table 1, Co-prompt consistently shows a robust performance gain in all scenarios, regardless of LLM, the dataset, and the retriever. Specifically, Co-Prompt, applied to OPT, achieves better results than the other methods. This indicates that the prompts generated by our method are more appropriate to play the role of an instruction to guide LLMs against other prompt optimization methods. More detailed results of re-ranked performance with various metrics are shown in Appendix B.3.

**Impact of Start Tokens** We exploit other options of start token such as "Score" and "This" as shown in Table 2. Regardless of the start tokens, Co-Prompt consistently generates prompts eliciting the performance of LLM efficiently. However, we observe that finding the optimal start token for the dataset is important to achieve better results.

**Impact of Generator** As shown in Table 3, even if different generators are used, the generated prompts by different generators guide the zero-shot re-ranker efficiently. Still, the differences in performance are caused by a vocabulary mismatch between the two modules. We see that, although our method does not vary significantly in performance to the generator, a more suitable generator may be necessary for better results.

**Relevance Score** We analyze the distributions of relevance scores between positive or negative document-query pairs. As the negative documents for a given query are retrieved from BM25, the negative ones are related to the query but unable to directly find the answer. As shown in Figure 2, we point out that the distribution difference exists between pairs despite some overlap. Also, an LLM

[1] http://beir.ai/

[2] https://huggingface.co/models

can distinguish which pair is positive, even without a prompt. However, we observe that the effect of discrete prompt optimization on the zero-shot re-ranker is in the direction of increasing the mean and variance of the relevance score.

**Case Study of Prompts** Table 2 shows the discrete prompts generated by our method and discrete prompt baselines when exploiting OPT as a re-ranker. While the prompts from the RL-prompt are ungrammatical gibberish close to a random word sequence, our method, Co-Prompt, generates interpretable prompts for humans, following human language patterns, and surpasses the performance of the other discrete prompts. Also, the word '*question*', one of the keywords describing the task, is included in the prompts from Co-Prompt regardless of the datasets. This implies that the prompts from our method can provide a natural user interface to improve human understanding of how LLMs work. See Appendix B.3 for more examples of Co-Prompt.

## 6 Conclusion

In this paper, we propose Co-Prompt, left-to-right prompt optimization for zero-shot re-ranker via constrained generation. Co-Prompt effectively restricts prompt candidates and evaluates the optimum of these prompts without any parameter updates. We experimentally show that our method achieves consistently outperforming performance across all experiments. Also, the impact of prompt optimization including baselines on the zero-shot re-ranker highlights its importance. We also present an interesting outcome in that the optimal prompt is interpretable for human. For future work, we plan to expand our method to other open-ended generation tasks using LLMs.

## Limitations

As shown in Table 1, our method is experimentally demonstrated to be effective for two LLMs. However, OPT, a decoder-only model, is more suitable for the prompts generated by Co-Prompt. This seems to be because T0, the encoder-decoder model, requires a separate generator such as GPT-2. The performance of prompts may vary to the generator involved in the vocabulary and training process. Also, there is a trade-off between search time and performance. While increasing the beam size and the number of document-query pairs enhances the probability of finding a more optimal prompt, it makes the search time proportionally longer.

## Ethics Statement

Our work contributes to enhancing the retrieval performance of a zero-shot re-ranker by optimizing the discrete prompt via constrained generation. We are keenly aware of the possibility of offensive or upsetting prompts caused by bias of the generator itself even though there were no such prompts in our experiments. Because there is no additional training for prompt optimization, our method has difficulty removing the bias of the language model itself. As studies on reducing the bias of language models or filtering out inappropriate expressions in texts are being actively conducted, these problems are expected to be sufficiently resolved in the future.

## Acknoledgements

This work was supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (No. 2018-0-00582, Prediction and augmentation of the credibility distribution via linguistic analysis and automated evidence document collection).

## References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Antoine Chaffin, Vincent Claveau, and Ewa Kijak. 2022. PPL-MCTS: Constrained textual generation through discriminator-guided MCTS decoding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2953–2967, Seattle, United States. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models:

A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Jia-Huei Ju, Jheng-Hong Yang, and Chuan-Ju Wang. 2021. Text-to-text multi-view learning for passage re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1803–1807, New York, NY, USA. Association for Computing Machinery.

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

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

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. In *CoCo@NIPS*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Cicero Nogueira dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Beyond [CLS] through ranking by generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1722–1727, Online. Association for Computational Linguistics.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2022. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and

Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

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):9.

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(140):1–67.

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

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

Devendra Singh Sachan, Mike Lewis, Dani Yogatama, Luke Zettlemoyer, Joelle Pineau, and Manzil Zaheer. 2023. Questions are all you need to train a dense passage retriever.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

# A Implementation Details

## A.1 Datasets

We employ two information retrieval datasets for evaluating the performance of the zero-shot re-ranker with the prompts. **1) MS-MARCO** (Nguyen et al., 2016) contains about 8M passages and 6,980 queries in development split collected from the Bing search logs. Because of the diversity of topics and contents with the large training set, recent work exploits MS-MARCO for retriever training (Nogueira and Cho, 2019; Qu et al., 2021). **2) Natural Question (NQ,** Kwiatkowski et al. (2019)) contains about 2M passages of Wikipedia articles and 3,452 queries in test split collected from Google search engines. Also, NQ, one of the popular open-domain question datasets, is exploited as training data of dense retrievers (Karpukhin et al., 2020). Both datasets are the benchmarks for evaluating information retriever systems (Thakur et al., 2021). Only 1,500 document-query pairs from MS-MARCO test split and NQ development split each are utilized for the prompt optimization.

## A.2 Metrics

As mentioned in Section 4, we employ two metrics, **1) ACC** and **2) nDCG**. In addition, we use one more metric. **3) MAP** is the mean average precision of the relevant documents' ranks for a given query.

## A.3 Retrievers

We use two types of retrievers, sparse and dense retrievers, for retrieving documents re-ranked by LLMs. **1) BM25** (Robertson and Zaragoza, 2009) is a representative sparse retriever computing the relevance score between a document and a query based on term frequency and inverse document frequency. BM25 has been widely employed because of its fast speed and effective performance. **2) DPR** (Karpukhin et al., 2020) interprets training dense retrieval as metric learning problems. The bi-encoder initialized with BERT (Devlin et al., 2019) is trained with contrastive learning exploiting positive and negative passages for a given query. It shows outperforming results over traditional sparse retrievers.

## A.4 Zero-shot Re-rankers

We employ two LLMs, T0 and OPT, as re-rankers with the prompt. **1) T0**, one of the T5 series (Raffel et al., 2020), consists of transformer encoder-decoder layers. The models are fine-tuned versions of T5 for multi-task learning with prompted datasets. **2) OPT**, a publicly open model, consists of decoder-only transformer layers. Its performance is comparable to those of GPT-3 models. We exploit OPT instead of GPT-3 due to academic budget.

The template is needed when trasmitting a document, a prompt and a query to zero-shot re-ranker together. Following the template setting of UPR, the template used in the experiments is "Passage: {document} {delimiter} {prompt} {delimiter} {query}". The delimiters used in the experiments are " " for T0 and "\n" for OPT.

## A.5 Baselines

**Manual Prompt** Sachan et al. (2022) not only proposed unsupervised passage re-ranker exploiting LLMs but also carefully selected the optimal prompt among handcrafted candidates validated by the re-ranked result at BM25 passages of NQ development set. The manually optimized prompt "*Please write a question based on this passage*" effectively guides zero-shot re-rankers to generate the query corresponding to the document.

**P-tuning** Liu et al. (2021) proposed P-tuning[3], generating soft prompts (i.e., continuous prompt embeddings), not discrete ones. They employed the prompt encoder consisting of long-short term memory layers trained to return the optimal soft prompts for the task. While the method mainly focuses on the text classification task, we define the loss objective as query generation log-likelihood for application to re-ranking. The prompt encoder is trained with document-query pairs for 10 epochs to generate 10-length soft prompts.

**RL-Prompt** Deng et al. (2022) proposed discrete prompt generation, applicable to open-ended generation tasks, with reinforcement learning. They validated the method applicable to text style transfer, one of open-ended text generation techniques. In order to align to the re-ranking task, we define the reward for the policy network as query generation log-likelihood from the document and the prompt. Following the setting mentioned in RL-Prompt[4], the 5-token length prompt is created through 12,000 training steps with a policy network model.

[3] https://github.com/THUDM/P-tuning

[4] https://github.com/mingkaid/rl-prompt

| Retriever \Re-ranker | Prompt \Generator | Instruction Prompt | MS-MARCO nDCG@20 | nDCG@100 | MAP@20 | MAP@100 |
|---|---|---|---|---|---|---|
| BM25 | - | - | 22.84 | 28.70 | 18.69 | 65.78 |
| T0 | Manual Prompt | "*Please write a question based on this passage.*" | 30.31 | 36.13 | 24.03 | 25.22 |
| | GPT2-Base | "*Please and tell me why, what, how,*" | 30.76 | 36.44 | 24.54 | 25.70 |
| | GPT2-Large | "*Please send me some info on why or in detail*" | **31.11** | **36.79** | **24.82** | **25.99** |
| | GPT2-XL | "*Please enter the message content, such\n and\n*" | 29.86 | 35.71 | 23.99 | 25.17 |

Table 4: Comparison of the prompts from the different generators and evaluation on the document set retrieved from MS-MARCO by BM25. The best results of each metric are marked in **bold**.

| Retriever \Re-ranker | | NQ ACC@20 | NQ ACC@100 | MS-MARCO ACC@20 | MS-MARCO ACC@100 |
|---|---|---|---|---|---|
| BM25 | | 62.9 | 78.3 | 48.0 | 66.7 |
| T0 | + Base Metric | 75.0 | 83.8 | 61.9 | 78.0 |
| | + Contrastive Metric | 76.2 | 83.8 | 59.6 | 76.2 |
| OPT | + Base Metric | 75.2 | 84.1 | 59.3 | 77.2 |
| | + Contrastive Metric | 74.4 | 84.0 | 57.7 | 75.7 |
| DPR | | 79.2 | 85.7 | 37.5 | 55.5 |
| T0 | + Base Metric | 80.4 | 87.7 | 58.0 | 71.7 |
| | + Contrastive Metric | 80.6 | 87.9 | 56.4 | 70.8 |
| OPT | + Base Metric | 80.2 | 88.1 | 56.4 | 71.3 |
| | + Contrastive Metric | 80.2 | 87.9 | 53.3 | 68.9 |

Table 5: Comparison between two options of likelihood at the ACC-$k$ accuracy.

# B Analysis

## B.1 Likelihood $P_{M_D}(D_s|\rho_{1:t})$

In this section, we call the likelihood proposed in Equation 4 as the base metric. We consider the other option of likelihood $P_{M_D}(D_s|\rho_{1:t})$ in a contrastive manner and also show the compared result with base metric in Table 5.

**Contrastive Measurement** The query generation score should be high for positive document-query pairs $D_s^+$ and low for negative pairs $D_s^-$. In a contrastive manner, the likelihood exploits the contrast between $P_{base}(D_s^+|\rho)$ and $P_{base}(D_s^-|\rho)$ as follows:

$$P_{cont}(D_s|\rho) = \frac{P_{base}(D_s^+|\rho)}{P_{base}(D_s^+|\rho) + P_{base}(D_s^-|\rho)} \quad (5)$$

As shown in Table 5, base metric gains a certain level of performance regardless of the dataset and LLM, whereas contrastive metric shows inferior performance over MS-MARCO.

## B.2 Impact of Generator

We show more detailed results of the prompts from the different generators in table 4. While the generated prompts follow human language patterns, there are some differences in used words.

## B.3 Detailed Results

We evaluate the performance of zero-shot re-ranker with various metrics at Top-20 and Top-100 documents, as shown in Table 6. Co-Prompt is ranked 1st or 2nd on every metric across all experiments. On the other hand, the manual prompt, optimized for NQ, records inferior performance over MS-MARCO. Also, other optimization methods, RL-Prompt and P-Tuning, fail to achieve the best record in all experiments. This shows that the optimal prompt for zero-shot re-ranker is made from our method, Co-Prompt.

In addition, when confirming qualitatively generated prompts, the outputs from Co-Prompt are similar to human language patterns compared to RL-Prompt. The keyword "*question*" is included in most of the prompts generated by Co-Prompt. Considering that other optimization methods produce dense prompt embedding or ungrammatical gibberish, Co-Prompt suggests a new direction in which a prompt can function as a natural user interface to understand a black-box model.

| Retriever \Re-ranker | | Instruction Prompt | NQ ACC@20 | ACC@100 | nDCG@20 | nDCG@100 | MAP@20 | MAP@100 |
|---|---|---|---|---|---|---|---|---|
| BM25 | | - | 62.9 | 78.3 | 20.2 | 23.9 | 7.8 | 9.9 |
| T0 | Null | "" | 73.1 | 82.8 | 27.8 | 32.1 | 12.9 | 16.0 |
| | P-Tuning | - | 72.9 | 82.8 | 27.9 | 32.2 | 12.8 | 16.0 |
| | RL-Prompt | " *poll question question question knows*" | 74.7 | 83.4 | 30.4 | 34.6 | 14.4 | 17.9 |
| | Manual | "*Please write a question based on this passage*" | **75.7** | **83.8** | **32.5** | **36.6** | **15.9** | **19.7** |
| | | "*Please try and find out the answer by asking questions like*" | 75.0 | **83.8** | 30.9 | 35.1 | 14.8 | 18.4 |
| | Co-Prompt | "*Please try and find out the answer by asking questions below*" | 75.1 | 83.5 | 31.0 | 35.2 | 15.0 | 18.5 |
| | | "*Please try and find out the answer by asking questions* | 75.3 | 83.7 | 31.0 | 35.1 | 14.9 | 18.4 |
| OPT | Null | "" | 70.5 | 81.9 | 25.1 | 29.8 | 11.1 | 14.0 |
| | P-Tuning | - | 71.2 | 82.9 | 27.2 | 32.1 | 12.5 | 15.9 |
| | RL-Prompt | " *poll trivia trivia wondered asking*" | 72.5 | 82.9 | 27.2 | 31.7 | 12.3 | 15.5 |
| | Manual | "*Please write a question based on this passage*" | 73.2 | 83.3 | 27.9 | 32.5 | 12.9 | 16.2 |
| | | "*Please post your question again when its not about the*" | 75.2 | **84.1** | 30.4 | 34.9 | 14.4 | 17.9 |
| | Co-Prompt | "*Please post your question again when its not just about*" | **75.5** | **84.1** | **30.6** | **35.1** | **14.7** | **18.2** |
| | | "*Please post your question again after doing research about* | 74.5 | 83.9 | 29.6 | 34.2 | 14.0 | 17.4 |
| DPR | | - | 79.2 | 85.7 | 34.0 | 35.2 | 17.9 | 19.8 |
| T0 | Null | "" | 78.5 | 86.6 | 31.4 | 34.9 | 15.9 | 18.6 |
| | P-Tuning | - | 79.1 | 87.0 | 32.1 | 35.4 | 16.1 | 19.0 |
| | RL-Prompt | " *poll question question question knows*" | 79.9 | 87.4 | 34.1 | 37.5 | 17.4 | 20.5 |
| | Manual | "*Please write a question based on this passage*" | **81.4** | **87.8** | **36.6** | **39.7** | **19.1** | **22.5** |
| | | "*Please try and find out the answer by asking questions like*" | 80.4 | 87.7 | 34.5 | 38.0 | 17.8 | 21.0 |
| | Co-Prompt | "*Please try and find out the answer by asking questions below*" | 80.2 | 87.6 | 34.8 | 38.2 | 17.9 | 21.2 |
| | | "*Please try and find out the answer by asking questions* | 80.2 | 87.6 | 34.8 | 38.1 | 17.9 | 21.1 |
| OPT | Null | "" | 76.3 | 86.1 | 28.8 | 32.8 | 13.8 | 16.5 |
| | P-Tuning | - | 78.2 | 87.5 | 31.8 | 36.1 | 15.9 | 19.2 |
| | Manual | "*Please write a question based on this passage*" | 78.9 | 87.5 | 32.0 | 35.8 | 16.0 | 19.0 |
| | RL-Prompt | " *poll trivia trivia wondered asking*" | 79.1 | 87.0 | 31.6 | 35.2 | 15.7 | 18.6 |
| | Manual | "*Please write a question based on this passage*" | 78.9 | 87.5 | 32.0 | 35.8 | 16.0 | 19.0 |
| | | "*Please post your question again when its not about the*" | **80.2** | 88.1 | 34.1 | 37.8 | 17.3 | 20.5 |
| | Co-Prompt | "*Please post your question again when its not just about*" | **80.2** | 88.0 | **34.6** | **38.2** | **17.8** | **21.0** |
| | | "*Please post your question again after doing research about* | 80.1 | **88.3** | 33.6 | 37.6 | 17.1 | 20.3 |

| Retriever \Re-ranker | | Instruction Prompt | MS-MARCO ACC@20 | ACC@100 | nDCG@20 | nDCG@100 | MAP@20 | MAP@100 |
|---|---|---|---|---|---|---|---|---|
| BM25 | | - | 48.0 | 66.7 | 25.2 | 28.7 | 18.7 | 19.2 |
| T0 | Null | "" | 53.2 | 72.7 | 27.5 | 31.2 | 20.2 | 20.7 |
| | P-Tuning | - | 54.1 | 72.5 | 28.5 | 31.9 | 21.1 | 21.6 |
| | RL-Prompt | " *question meaning difference meaning reality*" | 60.9 | 77.4 | 33.1 | 35.2 | 25.1 | 25.4 |
| | Manual | "*Please write a question based on this passage*" | 60.6 | 77.9 | 32.8 | 36.1 | 24.8 | 25.2 |
| | | "*Please send me some info on why or in detail,*" | **61.9** | **78.0** | **33.7** | **36.8** | **25.5** | **26.0** |
| | Co-Prompt | "*Please send me some info on why or in detail about*" | 61.2 | 77.8 | 33.4 | 36.6 | 25.4 | 25.9 |
| | | "*Please send me some info on why or in detail on*" | 61.2 | 77.7 | 33.3 | 36.5 | 25.2 | 25.7 |
| OPT | Null | "" | 50.4 | 71.7 | 25.4 | 29.4 | 18.3 | 18.8 |
| | P-Tuning | - | 56.4 | 75.5 | 29.4 | 33.0 | 21.6 | 22.1 |
| | RL-Prompt | "*questions answers key question defining*" | 59.2 | 76.7 | 31.5 | 34.8 | 23.4 | 23.9 |
| | Manual | "*Please write a question based on this passage*" | 55.3 | 74.6 | 28.7 | 32.4 | 21.1 | 21.6 |
| | | "*Please tell that is the first question asked on Google for*" | **59.3** | **77.2** | **31.9** | **35.2** | **23.9** | **24.4** |
| | Co-Prompt | "*Please tell that is the question of*" | 58.8 | 76.7 | 31.2 | 34.6 | 23.2 | 23.7 |
| | | "*Please tell that is the first question to arise on*" | 58.3 | 76.0 | 31.0 | 34.3 | 23.1 | 23.5 |
| DPR | | - | 37.5 | 55.4 | 19.6 | 22.9 | 14.6 | 15.0 |
| T0 | Null | "" | 51.5 | 68.0 | 27.8 | 30.9 | 20.9 | 21.3 |
| | P-Tuning | - | 52.5 | 68.2 | 28.5 | 31.5 | 21.6 | 22.0 |
| | RL-Prompt | " *question meaning difference meaning reality*" | 57.1 | 71.2 | 32.1 | 34.7 | 24.8 | 25.2 |
| | Manual | "*Please write a question based on this passage*" | 57.7 | **72.0** | 32.2 | 34.9 | 24.8 | 21.4 |
| | | "*Please send me some info on why or in detail,*" | **58.0** | 71.7 | **32.7** | **35.3** | **25.3** | **25.7** |
| | Co-Prompt | "*Please send me some info on why or in detail about*" | 57.6 | 71.6 | 32.5 | 35.1 | 25.1 | 25.5 |
| | | "*Please send me some info on why or in detail on*" | 57.3 | 71.6 | 32.3 | 35.1 | 25.1 | 25.5 |
| OPT | Null | "" | 50.1 | 68.1 | 26.4 | 29.7 | 19.5 | 20.0 |
| | P-Tuning | - | 54.6 | 69.9 | 29.1 | 32.0 | 21.8 | 22.2 |
| | RL-Prompt | "*questions answers key question defining*" | 56.3 | 71.1 | 31.1 | 33.8 | 23.7 | 24.1 |
| | Manual | "*Please write a question based on this passage*" | 54.3 | 70.1 | 29.1 | 32.1 | 21.9 | 22.3 |
| | | "*Please tell that is the first question asked on Google for*" | **56.4** | **71.3** | **31.4** | **34.2** | **24.1** | **24.5** |
| | Co-Prompt | "*Please tell that is the question of*" | 56.3 | 71.1 | 31.1 | 33.9 | 23.7 | 24.1 |
| | | "*Please tell that is the first question to arise on*" | 55.8 | 70.6 | 30.8 | 33.5 | 23.5 | 23.9 |

Table 6: Detailed results of LLM re-ranker with different prompts. The performance is evaluated with the three metrics at top-20 and top-100 documents.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*I discuss the limtations at section "Limitation".*

☑ A2. Did you discuss any potential risks of your work?
*I discuss the potential risk at section "Ethics Statement".*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*I write at section "Abstract" and 1.*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*Section 4 & Appendix A*

☑ B1. Did you cite the creators of artifacts you used?
*Section 4 & Appendix A*

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*Section 4 & Appendix A*

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Section 4 & Appendix A*

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Section 4 & Appendix A*

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Section 4 & Appendix A*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Section 4 & Appendix A*

### C ☑ Did you run computational experiments?

*section 5 Result*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Appendix A*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Section 4 & Appendix A*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*section 5 Result*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Section 4 & Appendix A*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*No response.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*No response.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*No response.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*No response.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*No response.*