Towards Incorporating Personalized Context for Conversational Information Seeking

Haitao Yu[1,*], Lingzhen Zheng[2], Kaiyu Yang[2], Sumio Fujita[3] and Hideo Joho[1]

[1]Institute of Library, Information and Media Science, University of Tsukuba, Tsukuba City, Ibaraki, Japan
[2]Graduate School of Comprehensive Human Sciences, University of Tsukuba, Tsukuba City, Ibaraki, Japan
[3]LY Research, LY Corporation, Tokyo, Japan

Abstract

Conversational information seeking (CIS) extends the classic search to a conversational nature, which has attracted significant attention in recent years. Yet one size does not fit all, it is no surprise that users often need high-quality personalized response due to their different personas, e.g., for the search about *alternatives to cow's milk*, the desired responses may differ a lot. In this work, we focus on CIS that aims to account for *personalized retrieval and response generation*. Specifically, we follow the CIS paradigm presented in the TREC iKAT track, which consists of three core tasks, namely *personal textual knowledge base (PTKB) statement ranking*, *passage ranking*, and *response generation*. For PTKB statement ranking, we propose to fuse multiple large language models (LLMs). For passage ranking, we propose four different strategies for personalized retrieval. For response generation, we resort to zero-short LLM-based answer generation by incorporating personalized context. The experimental results show that: (1) For PTKB statement ranking, our method achieves the best performance in terms of MRR on the set of iKAT organizers' assessments. It also shows superior performance over the baseline based on GPT-4. This indicates that a fusion of multiple LLMs is a promising choice when tackling problems of this kind. (2) For passage ranking, on one hand, one of our proposed strategies is able to achieve comparable performance as Llama2-based baseline. On the other hand, our analysis indicates that the way of incorporating PTKB statements for personalized retrieval matters, where a direct concatenation is not recommended. (3) For response generation, our proposed method is able to generate grounded and natural personalized responses, and is comparable to the top-tier LLM-based baseline.

Keywords

Conversational, Information Seeking, Personalized Context, LLM

1. Introduction

In recent years, *conversational systems* have attracted considerable attention from both academic researchers and industrial practitioners. In the field of information retrieval (IR), *conversational information seeking* (CIS) has been identified as one of the most important research directions. Remarkable efforts have been made from different aspects, which include, but not limited to, *conversational search conceptualization* [1, 2, 3], *conversational query re-writing* [4, 5, 6], *generating and selecting clarifying questions* [7, 8, 9, 10] and *conversational response generation* [11, 12, 13].

Despite the successes achieved by the aforementioned studies, fundamental research questions remain open. For example, providing high-quality user-specific response is still a challenging problem. Take the case by Aliannejadi et al. [14] as an example, for the search about alternatives to cow's milk, two personas can be: (A) *Alice is a vegan who is deeply concerned about the environment*; and (B) *Bob has been recently diagnosed with diabetes, has a nut allergy, and is lactose intolerant.* Given Alice and Bob's personas, their corresponding conversations with the system would evolve and develop in very different ways. Put another way, the responses that are helpful to Alice may not be necessarily useful to Bob, and vice versa. In fact, information needs of this kind are prevalent in daily information searches, which include, but not limited to, job finding, healthcare search and online shopping. Given the information needs expressed as a sequence of search queries (or questions) and different personas, it is of great importance that the CIS system can effectively incorporate the personalized context and provide relevant responses to users. Motivated by this observation, we focus on developing a unified CIS system, which enables to incorporate personalized context during the interactive search process. The main contributions of this work are listed as follows:

- By following the CIS paradigm presented in the TREC iKAT track, we propose different methods for tackling the core tasks, namely *personal textual knowledge base (PTKB) statement ranking*, *passage ranking*, and *response generation*. For PTKB statement ranking, we explore how to fuse multiple large language models (LLMs). The experimental results show that our method achieves the best performance in terms of MRR on the set of iKAT organizers' assessments which relies on a larger assessment pool. Moreover, our method also shows superior performance over the GPT-4-based baseline. This highlights that it is not straightforward to solve a component task by merely tailoring a powerful LLM. Whereas a fusion of multiple LLMs can be a promising choice when tackling problems of this kind.
- For passage ranking, we propose four different strategies for personalized retrieval, which enables us to well investigate the impact of utterance rewriting and the way of incorporating personalized context. Through result analysis and comparison, we found that: Though our proposed method for selecting PTKB statements is relatively reliable, how to incorporate the selected PTKB statements to formulate the input for personalized retrieval matters a lot. A direct concatenation is not suggested according to the inferior performance of our proposed strategies.
- For response generation, we resort to zero-short LLM-based answer generation by incorporating personalized context. Our method is able to generate

Information Retrieval's Role in RAG Systems (IR-RAG), 18 July, 2024, Washington, DC
*The corresponding author.
yuhaitao@slis.tsukuba.ac.jp (H. Yu); s2221686@u.tsukuba.ac.jp (L. Zheng); s2321730@u.tsukuba.ac.jp (K. Yang); sufujita@lycorp.co.jp (S. Fujita); hideo@slis.tsukuba.ac.jp (H. Joho)
0000-0002-1569-8507 (H. Yu); 0009-0004-5783-7079 (L. Zheng); 0009-0002-4491-7235 (K. Yang); 0000-0002-1282-386X (S. Fujita); 0000-0002-6611-652X (H. Joho)

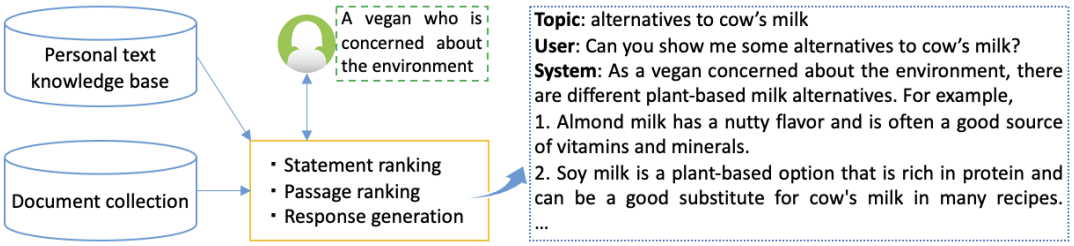

Figure 1: Our focused framework for conversational information seeking that incorporates personalized context.

grounded and natural personalized responses, and is comparable to the top-tier LLM-based baseline.

2. Preliminaries

Figure 1 describes our focused framework for CIS that accounts for users' personas. It assumes that there is a personal text knowledge base (PTKB), which consists narrative sentences providing personal information about the users. A system following this framework consists of the following key modules. (1) *Statement ranking*: given the context of the conversation and the current user utterance, this module returns a ranked list of PTKB statements based on their relevance, which reflects the user's persona; (2) *Passage ranking*: given the context of the conversation, the current user utterance, and the PTKB statements, this module is responsible for retrieving a ranked list of passages from the document collection; (3) *Response generation*: this module returns the answer text as a response to the user. In particular, the response should be a generative or abstractive summary of the relevant passages. We recognize that the gap exists between our focused framework for CIS and the real-world search scenarios. Since this topic is still in its infancy, we leave it as a future work to explore more complex frameworks.

3. Methodology

Given the target paradigm for CIS in section 2, we elaborate on the proposed methods for addressing the key module as below.

3.1. Statement Ranking by Fusing Multiple LLMs

The key idea of our method (denoted as SR_FML) for tackling statement ranking is to effectively fuse multiple LLMs through a cascade of four steps. At the first step, we rewrite each conversation turn's utterance. Specifically, the T5-CANARD model [15] fine-tuned with the testing topics of TREC CAsT 2022 [16] is used, and the preceding turns' conversations (3 turns at most) are used as the context. At the second step, given the candidate PTKB statements, we perform binary logistic regression based on the BERT [17] model. The candidate PTKB statements with a true label are kept for later steps, and the statements with a false label are filtered out. At the third step, we perform binary logistic regression again over the remaining PTKB statements based on MonoT5 [18] in the same way as the second step. In addition, we use RankGPT [19] to sort the PTKB statements, and assign the top half statements with a true label, and a false label for the remaining bottom statements. At the fourth step, we mange to unify the ranking information and binary classification results of the previous two steps via a scoring function and an indicator function. The scoring function assigns a weight for each remaining statement in the 2nd step as follows:

$$w(s) = 1 - \frac{Ind_{MonoT5}(s) + Ind_{RankGPT}(s)}{2 * |S|} \quad (1)$$

where $Ind_{RankGPT}(s)$ and $Ind_{MonoT5}(s)$ represent the rank positions according to the regression scores by MonoT5 and RankGPT, respectively. $|S|$ represents the number of remaining PTKB statements in the second step. The indicator function builds upon $w(s)$ and a voting mechanism as follows:

$$I(s) = \begin{cases} 1 & \text{if } (lab_{BERT}(s) + lab_{MonoT5}(s) + lab_{RankGPT}(s)) \geq 2 \\ & \quad \text{and } w(s) > 0.65 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$lab_{BERT}(s)$, $lab_{MonoT5}(s)$, and $lab_{RankGPT}(s)$ respectively represent the binary classification result by each adopted LLM, where an output of 1 denotes a true label, and 0 for a false label.

The final result list of PTKB statements is generated by selecting statements with a positive output via the indicator function and ranking them via the scoring function in a decreasing order.

3.2. Zero-shot LLM-based Passage Ranking

To cope with passage ranking, we resort to the typical pipeline of retrieve-then-rank. Firstly, we use BM25 with the default setting in Pyserini to retrieve the top 5 passages. Then we design 4 strategies (denoted as PR_S1, PR_S2, PR_S3 and PR_S4, respectively) to re-rank the top 5 passages using multiple specifically selected LLMs in a zero-shot manner.

To formulate the input, PR_S1, PR_S3, and PR_S4 concatenate the rewritten utterance and the top 2 relevant PTKB statements returned by the module of statement ranking. PR_S2 directly uses the rewritten utterance as the input.

During the ranking process, the differences among the four strategies are as follows: (1) PR_S1 and PR_S2 assemble the results by multiple LLMs (i.e., *"stabilityai/stablelm-tuned-alpha-7b", "eachadea/vicuna-13b-1.1", "jondurbin/airoboros-7b", "TheBloke/koala-13B-HF"*) [20, 21, 22, 23, 24] in a voting manner. Specifically, given the information need represented by the input, we ask each LLM to compare the candidate passages in a pairwise manner. The passage that is identified to be more relevant than the other gets a vote. Finally, we rank the

passages based on the cumulative number of votes in a decreasing order; (3) PR_S3 merely relies on MonoT5 with the default setting in PyGaggle to rank the passages; (4) PR_S4 relies on the idea of RankGPT to rank the passages, where the GPT-3.5 API is used.

3.3. Personalized Response Generation

For tackling response generation, we aim to generate personalized response. Specifically, for each conversation turn, the top-1 passage and the top-2 PTKB statements representing the personalized context are used as the input. For the base LLM, we resort to T5 [25], which is specifically fine-tuned for the summarization task.

4. Experimental Setup

4.1. Dataset

We use the dataset released by TREC iKAT 2023 for evaluating the effectiveness with 25 testing topics. Each topic has $1 \sim 3$ subtree conversations that represent different personas. For each personalized conversation, there is a list of around 10 PTKB statements. Moreover, the passage collection has 116, 838, 987 passages, which is derived from a subset of ClueWeb22-B [26].

4.2. Baselines

In order to make a fair and thorough analysis, we perform a module-specific comparison by selecting the most competitive and representative baseline methods from TREC iKAT 2023's participants. We add a prefix of *BS* to each baseline method for a better clarity.

For statement ranking, BS_zs_Llama and BS_ft_Llama use zero-shot and fine-tuned Llama-2-7b-chat [27] for rewriting the utterance, respectively. Then they use MiniLM12 [28] to rank PTKB statements based on the rewritten utterance.

For passage ranking, BS_Llama2 initially instructs Llama-2-7b-chat to reformulate the current utterance considering previous conversation turns' context. Then, the revised conversation, along with a specific passage, are provided to the model to assess the passage's relevance.

For response generation, BS_FastChatT5andLlama creates a summarization for each of the top passages retrieved by BM25 using FastChatT5 [29], then it generates the response to current utterance based on the summaries in a retrieval-generate loop. A final response is summarized by BS_DenseMonoT5 using different engines including conventional language models and Llama2 based on top passages.

Besides the above module-specific baseline methods, BS_GPT-4 is compared across three modules, which represents the method using the most powerful LLM (i.e., GPT-4 [30]). For statement ranking, BS_GPT-4 casts it as a binary classification problem. The prompt includes the instruction, context of the conversation, PTKB statements of the user, and current user utterance. The output is a ranked list of relevant statements. For passage ranking, BS_GPT-4 initially generates an answer for each turn. Subsequently, GPT-4 is employed to produce five queries for each answer. These generated queries are used via BM25 to retrieve passages, then the pre-trained MiniLM12 is deployed for ranking the passages. For response generation, GPT-4 is prompted to generate the answer, using the top-10 retrieved passages, the top-3 PTKB statements, the context of the conversation and the user utterance.

4.3. Implementation Details

All experiments were conducted on a server with two A100 (40GB) GPUs. The CUDA version is 12.2. For fine-tuning T5-CANARD, the configuration is: training epochs: 5, batch size: 4, learning rate: $1e-5$. For SR_FML, bert-base-uncased with default parameter settings is used as the backbone model, which comes from *transformers* library provided by HuggingFace [31]. We iterate its predictions five times and compute the average relevance scores for each statement. For RankGPT, the configuration is: window size: 4, step size: 1. The MonoT5 with default parameter settings in *Pygaggle* is used. In PR_S3, the window size of RankGPT is adjusted to 3. In PR_S1 and PR_S2, we set the prompt_max_length of the four zero-shot LLMs to 2048. Additionally, we set the decoding method to beam_search, output_max_length to 512, and temperature to 1.0 by default [32]. For RG_SumT5, t5-base-finetuned-summarize-news is employed with configuration: input max_length: 512, output min_length: 50, output max_length: 150, length_penalty: 2.0, num_beams: 4.

5. Results and Analysis

In Table 1, Table 2 and Table 3, we show the overall performance of the baseline approaches, and the proposed methods for statement ranking, passage ranking and response generation, respectively. Within each table, the best result in terms of each metric is indicated in bold, and the second-best result is underlined.

For statement ranking, we note that there are two sets of assessments which were created by the iKAT organizers and NIST assessors, respectively. The key differences are that: During topic generation, the organizers annotated each turn in terms of their provenance to PTKB statements and included their labels in the released topic files. During the assessment of passage relevance, the NIST assessors were also asked to judge the relevance of PTKB statements to each turn. The assessment pool is smaller than the one done by the organizers. The organizers judged all of the turns, while the NIST assessors only judged the turns that were selected for passage relevance [14]. From Table 1, we can observe that BS_zs_Llama outperforms the other methods in terms of nDCG@3, P@3 and Recall@3. Though BS_ft_Llama relies on the same LLM, its performance is impacted due to the rewritten utterances in a fine-tune setting. On the contrary, BS_GPT-4 relying on the powerful GPT-4 shows inferior performance across two sets of assessments. This indicates that the usage of GPT-4 for statement ranking is not straightforward, further exploration is needed for a better performance. Over the set of iKAT organizers' assessments, our proposed method (i.e., SR_FML) shows competitive performance as BS_zs_Llama, and achieves the best performance in terms of MRR. This indicates the benefit of fusing multiple LLMs, which enables us to leverage on the advantages of different LLMs. In view of the fact that the set of iKAT organizers' assessments bases on a larger assessment pool, it is reasonable to say that the evaluation over this set is more reliable.

For passage ranking, the results in Table 2 show that BS_GPT-4 significantly outperform BS_Llama2 and our pro-

Table 1
The performance comparison on statement ranking.

Ground Truth	Method	Metric			
		MRR	nDCG@3	P@3	Recall@3
iKAT organizers' assessment	BS_zs_Llama	0.6707	**0.6394**	**0.3810**	**0.7375**
	BS_GPT-4	0.6618	0.6288	0.3423	0.6888
	BS_ft_Llama	0.6617	0.6149	0.3542	0.6918
	SR_FML	**0.6890**	0.6370	0.3512	0.6903
NIST assessment	BS_zs_Llama	**0.7950**	**0.7254**	**0.4626**	**0.6964**
	BS_ft_Llama	0.7795	0.7102	0.4490	0.6796
	BS_GPT-4	0.7027	0.6174	0.3605	0.5833
	SR_FML	0.7112	0.6594	0.4184	0.6213

Table 2
The performance comparison on passage ranking.

Method	nDCG@3	nDCG@5	mAP
BS_GPT-4	**0.4382**	**0.4396**	**0.1759**
BS_Llama2	0.1389	0.1466	0.0376
PR_S2	0.1433	0.1469	0.0350
PR_S4	0.1130	0.1070	0.0224
PR_S3	0.1107	0.1062	0.0223
PR_S1	0.1086	0.1049	0.0222

posed methods by a large margin. This echoes the findings in prior studies [19, 33, 34, 35] which have shown the leading capability of GPT-4 in the passage ranking task. One probable reason is that the pipeline of *generate-retrieve-generate* adopted by BS_GPT-4 is more suitable for passage ranking than our adopted pipeline of *retrieve-generate*. Among our proposed strategies for passage ranking, PR_S2 shows the best performance, and also outperforms BS_Llama2. Compared with BS_Llama2, a possible reason for the inferior performance of the other three strategies is the way of formulating the input. We directly concatenate the utterance and related PTKB statements as the input, while BS_Llama2 rewrites the utterance with the statements using LLM. Another possible reason for our inferior performance is that we focus on the earlier positions and only re-rank the top-5 passages returned by BM25. As a result, this setting would become a bottleneck for us to get relevant passages given the limited retrieval ability of BM25.

Table 3
The result comparison on response generation.

Method	Groundedness	Naturalness
BS_GPT-4	**0.89 (65/8)**	**4.0**
BS_FastChatT5andLlama	0.67 (47/23)	3.684
BS_DenseMonoT5	0.51 (37/36)	2.808
RG_SumT5	0.67 (49/24)	2.9178

For response generation, the results are evaluated in terms of groudedness and naturalness. *Groudedness* measures whether the generated response can be attributed to the passages that it is supposed to be generated from. *Naturalness* measures the extent to which the response sounds humanlike, such as the general fluency and understandability of the generated response. GPT-4 is used to evaluate both the groundedness and naturalness of the responses in each turn. Finally, the mean of groundedness and naturalness over all turns is reported. From Table 3, we can observe that BS_GPT-4 again outperforms the other methods by a large margin. Our proposed method (i.e., RG_SumT5) outperforms BS_DenseMonoT5 and shows competitive performance as BS_FastChatT5andLlama.

It is noticeable that the evaluation results are likely to be somewhat biased towards BS_GPT-4, since the evaluation is conducted by GPT-4. We leave it as a future work to further test the effectiveness of these methods for response generation through human evaluation results.

A joint look across Table 1, Table 2 and Table 3 reveals that: First, we do not observe a clear correlation between statement ranking and passage ranking, which seems counterintuitive. For instance, though BS_GPT-4 shows inferior performance in statement ranking, it outperforms the other methods by a large margin in passage ranking. This counterintuitiveness may arise from a number of possible reasons, such as the strong zero-shot capability of GPT-4 and the precise understanding of persona information underlying selected PTKB statements. This is also worthy to be investigated as a future work. Second, for both personalized retrieval and response generation in the context of CIS, there is still a large room to improve the performance.

6. Conclusion

In this study, we focus on CIS that accounts for personalized retrieval and response generation. By following the CIS paradigm presented in the TREC iKAT track, we propose different methods to tackle three core tasks, namely personal textual knowledge base (PTKB) statement ranking, passage ranking and response generation. We have shown that fusing multiple LLMs is a promising way for addressing PTKB statement ranking. Also, our analysis indicates that an effective way of injecting the selected PTKB statements is quite important for personalized retrieval. Since conversational systems arise in a variety of applications, such as recommender systems and question answering, we believe that our work provides insights for developing conversational systems that account for personalized retrieval and response generation.

7. Acknowledgments

This research has been supported by JSPS KAKENHI Grant Number 19H04215.

References

[1] L. Azzopardi, M. Dubiel, M. Halvey, J. Dalton, Conceptualizing agent-human interactions during the conversational search process, in: The second international workshop on conversational approaches to information retrieval, 2018.

[2] Y. Deldjoo, J. R. Trippas, H. Zamani, Towards multimodal conversational information seeking, in: Proceedings of the 44th International ACM SIGIR conference on research and development in Information Retrieval, 2021, pp. 1577–1587.

[3] F. Radlinski, N. Craswell, A theoretical framework for conversational search, in: Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, CHIIR '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 117–126. URL: https://doi.org/10.1145/3020165.3020183. doi:10.1145/3020165.3020183.

[4] S. Yu, J. Liu, J. Yang, C. Xiong, P. Bennett, J. Gao, Z. Liu, Few-shot generative conversational query rewriting, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 1933–1936.

[5] S. Vakulenko, S. Longpre, Z. Tu, R. Anantha, Question rewriting for conversational question answering, in: Proceedings of the 14th ACM international conference on web search and data mining, 2021, pp. 355–363.

[6] S.-C. Lin, J.-H. Yang, R. Nogueira, M.-F. Tsai, C.-J. Wang, J. Lin, Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting, ACM Trans. Inf. Syst. 39 (2021). URL: https://doi.org/10.1145/3446426. doi:10.1145/3446426.

[7] M. Aliannejadi, H. Zamani, F. Crestani, W. B. Croft, Asking clarifying questions in open-domain information-seeking conversations, in: Proceedings of the 42nd international acm sigir conference on research and development in information retrieval, 2019, pp. 475–484.

[8] H. Zamani, S. Dumais, N. Craswell, P. Bennett, G. Lueck, Generating clarifying questions for information retrieval, in: Proceedings of The Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 418–428. URL: https://doi.org/10.1145/3366423.3380126. doi:10.1145/3366423.3380126.

[9] I. Sekulić, M. Aliannejadi, F. Crestani, Towards facet-driven generation of clarifying questions for conversational search, in: Proceedings of the 2021 ACM SIGIR international conference on theory of information retrieval, 2021, pp. 167–175.

[10] H. Zamani, B. Mitra, E. Chen, G. Lueck, F. Diaz, P. N. Bennett, N. Craswell, S. T. Dumais, Analyzing and learning from user interactions for search clarification, 2020. arXiv:2006.00166.

[11] K. Wang, J. Tian, R. Wang, X. Quan, J. Yu, Multi-domain dialogue acts and response co-generation, arXiv preprint arXiv:2004.12363 (2020).

[12] C. Ye, L. Liao, F. Feng, W. Ji, T.-S. Chua, Structured and natural responses co-generation for conversational search, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 155–164.

[13] X. Gu, K. M. Yoo, J.-W. Ha, Dialogbert: Discourse-aware response generation via learning to recover and rank utterances, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2021) 12911–12919. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17527. doi:10.1609/aaai.v35i14.17527.

[14] M. Aliannejadi, A. Zahra, C. Shubham, D. Jeffery, A. Leif, Trec ikat 2023: The interactive knowledge assistance track overview, in: Proceedings of the Thirty-Second Text REtrieval Conference (TREC 2023), 2024.

[15] S.-C. Lin, J.-H. Yang, R. Nogueira, M.-F. Tsai, C.-J. Wang, J. Lin, Conversational question reformulation via sequence-to-sequence architectures and pretrained language models, arXiv preprint arXiv:2004.01909 (2020).

[16] P. Owoicho, J. Dalton, M. Aliannejadi, L. Azzopardi, J. R. Trippas, S. Vakulenko, Trec cast 2022: Going beyond user ask and system retrieve with initiative and response generation, NIST Special Publication (2022) 500–338.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[18] R. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document ranking with a pretrained sequence-to-sequence model, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 708–718. URL: https://aclanthology.org/2020.findings-emnlp.63. doi:10.18653/v1/2020.findings-emnlp.63.

[19] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, Z. Ren, Is chatgpt good at search? investigating large language models as re-ranking agents, 2023. arXiv:2304.09542.

[20] X. Geng, A. Gudibande, H. Liu, E. Wallace, P. Abbeel, S. Levine, D. Song, Koala: A dialogue model for academic research, Blog post, 2023. URL: https://bair.berkeley.edu/blog/2023/04/03/koala/.

[21] Y. Anand, Z. Nussbaum, B. Duderstadt, B. Schmidt, A. Mulyar, Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo, https://github.com/nomic-ai/gpt4all, 2023.

[22] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: An instruction-following llama model, https://github.com/tatsu-lab/stanford_alpaca, 2023.

[23] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, E. P. Xing, Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/.

[24] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research 21 (2020) 5485–5551.

[26] A. Overwijk, C. Xiong, J. Callan, Clueweb22: 10 billion

web documents with rich information, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 3360–3362.

[27] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[28] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.

[29] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. arXiv:2306.05685.

[30] O. (2023), Gpt-4 technical report, 2023. arXiv:2303.08774.

[31] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Q. Liu, D. Schlangen (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6. doi:10.18653/v1/2020.emnlp-demos.6.

[32] D. Jiang, X. Ren, B. Y. Lin, Llm-blender: Ensembling large language models with pairwise ranking and generative fusion, arXiv preprint arXiv:2306.02561 (2023).

[33] R. Pradeep, S. Sharifymoghaddam, J. Lin, Rankvicuna: Zero-shot listwise document reranking with open-source large language models, 2023. arXiv:2309.15088.

[34] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Dou, J.-R. Wen, Large language models for information retrieval: A survey, 2024. arXiv:2308.07107.

[35] R. Tang, X. Zhang, X. Ma, J. Lin, F. Ture, Found in the middle: Permutation self-consistency improves listwise ranking in large language models, 2023. arXiv:2310.07712.