Can smaller LLMs be tutored by bigger LLMs to become better rankers?

AHMED IBRAHIM, Delft University of Technology, Netherlands
CHANDRAN NANDKUMAR, Delft University of Technology, Netherlands
LEE WEN XIN NOEL, Delft University of Technology, Netherlands
PATRICIA VANESSA SANTOSO, Delft University of Technology, Netherlands

This paper aims to evaluate a novel LLM-based ranking approach against popular approaches, to investigate the potential of improving the efficiency of descriptive query ranking. We evaluate a GPT3.5 finetuned on 50 reasoning-based rankings crafted by a GPT-4-Turbo model with access to the ground truth against a standard GPT3.5, BM25 and Embeddings-based ranking approach. To evaluate our models, we used the MS Marco dataset and the Mean Reciprocal Rank (MRR) as a metric to measure the effectiveness of the different approaches. While our results do not show any significant improvement in ranking performance, the study shows potential in incorporating fine-tuning techniques in LLM-based ranking models and discusses what these results could hold for the future in terms of search engine technologies and information retrieval. The code and fine-tuning data is provided here - https://github.com/cn0303/Information-Retrieval-Project

CCS Concepts: • **Information systems** → **Rank aggregation**; **Retrieval efficiency**; **Relevance assessment**.

Additional Key Words and Phrases: Ranking Algorithms, Deep Learning, Traditional Ranking Methods, GPT 3.5, MS Marco Dataset

ACM Reference Format:
Ahmed Ibrahim, Chandran Nandkumar, Lee Wen Xin Noel, and Patricia Vanessa Santoso. 2024. Can smaller LLMs be tutored by bigger LLMs to become better rankers?. In *IN4325-Q3-24: Information Retrieval, TU Deflt*. ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Recent advancements in information retrieval systems have highlighted the importance of efficient and effective ranking mechanisms. A main concern is how these algorithms organize and prioritize data arrays in response to complex user queries. Information retrieval has been traditionally dominated by algorithmic approaches, such as BM25, which emphasize efficiency and relevance the use of different data sources, including textual content, user interaction data, and other related attributes [17]. The evolution of ranking systems has seen an influence from the integration of machine learning techniques over the past decade [2, 7, 9]. In a more recent timeframe, deep learning models, like those based on embeddings, vector databases, language models like BERT and large language models such as GPT-3.5, have introduced new approaches to IR ranking that could potentially enhance performance in interpreting complex queries [18].

These deep learning models are increasingly being used for ranking in the information retrieval domain to aid in interpreting complex queries [4, 21]. While these advancements highlight the potential of neural network-based models in improving information ranking, there is a noticeable research gap in the application and optimization of large language models (LLMs) for ranking tasks. Much of the existing literature focuses on the use of pre-trained models without exploring the influence of direct finetuning of these models. Finetuning LLMs by training them with

Manuscript submitted to ACM

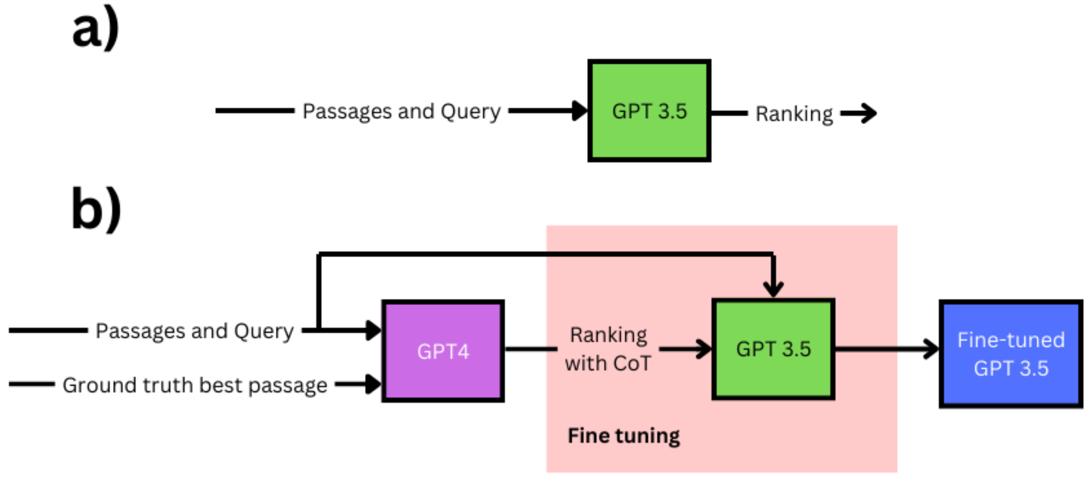

Fig. 1. Explanation of our proposed approach. While conventional LLMs take the relevant passages and query as input and output the ranking as seen in a), we propose using GPT4 with access to ground truth data to create Chain-Of-Thought reasoning-enabled explanations and use this to fine-tune a GPT3.5 model. This new model can then be used for ranking.

the intended data can probably influence the accuracy and applicability of these models for everyday IR tasks.

To bridge this gap, this research will compare between traditional information ranking methods, such as BM25, and more modern deep learning approaches, including a fine-tuned version of GPT-3.5. This research will explore the performance of a directly fine-tuned GPT-3.5 using a novel approach: using training data generated and explained by a larger and more advanced LLM, GPT-4, to see if this method can improve the ranking performance. This study aims to not only compare the effectiveness of traditional and neural network-based ranking algorithms in handling descriptive queries but also to investigate the potential performance improvements that can be achieved by fine-tuning LLMs with LLM-generated explanations. To achieve this, the following research question will be answered:

”Is it possible to improve the capabilities of a pre-trained Large Language Model (LLM) by fine-tuning it with Chain-Of-Thought reasoning outputs generated by a larger LLM? Moreover, how does this enhanced model’s ranking performance compare to traditional state-of-the-art information retrieval algorithms, such as BM25, embedding-based ranking systems, and the ranking capabilities of the original, unmodified pre-trained LLM?”

Sub-questions that will help answer this research question include:

(1) How does the MRR performance of the BM25 algorithm compare with neural network-based methods, including the baseline GPT-3.5 and an embeddings-based ranking approach, specifically for handling descriptive queries from the MS Marco dataset?

(2) What is the effect of fine-tuning GPT-3.5 with instruction from GPT4 using a selection of descriptive queries from the MS Marco dataset on its MRR performance, and how does this refined model perform relative to the original GPT-3.5 configuration?
(3) Considering the MRR outcomes obtained from the above comparisons, what conclusions can be drawn about the comparative strengths and limitations of each considered ranking algorithm in accurately ranking descriptive queries?

We hypothesise that *The fine-tuned GPT3.5 Turbo will outperform the GPT3.5 base model, BM25 and embeddings-based ranker for 50 descriptive test samples on the MS Marco test set*

The study finds that while fine-tuning GPT-3.5 appears to provide some improvement in ensuring the correct passage is ranked first, there appear to be no statistically significant differences in performance compared to the other models. We believe that further exploration and research must be done with larger training samples and with better fine-tuning approaches and prompt engineering to investigate if the results can be improved after a certain point.

2 RELATED WORK

BM25 is a widely used algorithm in the domain of traditional information retrieval algorithms [14]. It is known for its simplicity and efficacy in ranking documents based on their relevance to a query [15]. It originates from the probabilistic information retrieval model and calculates the relevance of documents to a query based on term frequency and inverse document frequency metrics. It also incorporates document length normalization to balance the bias towards longer documents [14]. Its performs well across different sorts of text-based data and is a preferred choice for systems where quick, reliable retrieval is needed such as web search engines [14]. BM25's algorithmic approach also has some drawbacks. It particularly lacks the complexity to understand a query intent or the semantic relationships between words, which could potentially limit its performance on complex or longer queries [11].

Embedding-based models on the other hand offer a more dynamic approach to ranking by representing words or phrases as vectors in a high dimensional space [19]. This allows capturing the semantic relationships based on their context within a document [1]. This method has advanced the field of ranking by enabling systems to 'understand' the intent of queries and documents. Embedding-based models, such as Word2Vec, GloVe, or BERT, have demonstrated better performance in ranking relevant documents for more complex queries [5, 13]. A drawback of using embedding-based models it their reliance on pre-trained models for generating embeddings. This can be a limitation for context-specific domains [10]. Compared to BM25, embedding-based approaches are computationally expensive and complex, but offer the ability to match queries and documents based on semantic similarity.

The emergence of large language models like GPT-3.5 has introduced a new ranking model. GPT-3.5, with its large knowledge database and ability to 'understand' complex language, has the potential to improve information retrieval tasks such as ranking [22]. Its application in information retrieval, through direct prompts or as part of larger ranking models, has shown promise in providing accurate responses to queries [16, 22]. Fine-tuning large language models such as GPT-3.5 on specific datasets or query types is expected to further improve its performance. Previous literature suggests that fine-tuning large language models for specific tasks can significantly improve the model's performance for those tasks [12]. The expectation is that a fine-tuned GPT-3.5 could outperform traditional and embeddings-based methods by leveraging its contextual understanding and generative capabilities. This will be explored further in this research.

Chain-of-Thought (CoT) [20] prompting is a technique that aims to induce Large Language Models (LLMs) with reasoning capabilities. By providing step-by-step instructions that mimic a person's thought process, the model learns to breakdown a problem into intermediate reasoning steps and arrive at the desired solution. CoT has led to improvements in the performance of LLMs over a wide range of tasks. Often this can be achieved with few-shot learning [20], where a handful of manually written prompts are fed into the model, or even zero-short learning [6].

Studies have also shown how small language models (SLMs), when trained with LLM-generated explanations, significantly outperform fine-tuning baselines and can even surpass the performance of larger LLM models[8]. Engaging this concept in a similar vein, fine-tuning GPT-3.5 with the superior reasoning and explanatory capabilities of GPT-4 would thus significantly improve our achieved ranking model, potentially outperforming its superior LLM counterparts with its new application of chain-of-thought reasoning in the context of information retrieval systems. This approach underscores the potential of deploying cost-effective, yet powerful, ranking models, bridging the gap between computational efficiency and advanced reasoning capabilities facilitated by superior, but more computationally expensive LLMs.

3 METHODOLOGY

This study has compared the performance of four different ranking algorithms on the MS Marco dataset[1] to understand their effectiveness in ranking descriptive queries. The 4 different ranking approaches evaluated in this study are - BM25, Embeddings based ranking, base GPT3.5 Turbo 0125 as a ranker and fine-tuned GPT3.5 Turbo 0125 as a ranker.

3.1 Dataset

The MS Marco dataset is a large and publicly available set of query and answer pairs. Each pair is sourced from actual user queries. The answers contain different search results, each result is then marked with either a 0 or a 1. The result that has been clicked by the user, which is the result that should be ranked first, is labelled with a 1. For this research, a subset of 50 descriptive queries was selected randomly for training and fine-tuning the LLM and 50 descriptive queries were used for testing from the test set.

3.2 Approach for finetuning GPT 3.5

The MS Marco dataset used comprises several potentially relevant passages per query and the ground truth as to which passage provides the most relevant answer to the query in the is_selected column. We provide a prompt for enabling the GPT3.5 to take in n number of passages and a query to present a result. Note, the part of the prompt enclosed inside <SYS> and </SYS> is the system prompt and the passages and query are obtained from variables outside the prompt each time.

[1] https://huggingface.co/datasets/ms_marco

```
<SYS>You are a ranker. Your job is to take a number of passages and a query as input and
    output an ordered list, comma separated with the passage numbers inside () of the
    correct ranked order of these items. For example - (6,1,3,4,2,5) </SYS>

<SYS> ALWAYS ENCLOSE THE FINAL LIST INSIDE () AND INSIDE THERE SHOULD ONLY BE DIGITS AND
    , IN THE RIGHT ORDER. </SYS>

Given the following passages -
<Passage 1>
<Passage 2>
...

and the query -
<query>

Provide a ranking of the different passages in the following format (best_passage_number,
     second_best, ...).
```

The given prompt along with the passages and query as input results in the LLM providing a complete ranking of all the passages. The only hyperparameter which was specified was the temperature which was set to *0.2*. This enables for more controlled and deterministic outputs rather than the default 0.6.

However, if we wish to use a powerful model like GPT4 to ’teach’ ranking to a smaller model, we need to provide GPT4 with the relevant passages, query and ground truth. We instruct GPT4 that this ground truth information must not be referenced anywhere in its response and that it is a secret provided only to it and not to the model it is supposed to teach. We also ask it to provide strong explanations to justify its ranking by using the ground truth. To ensure that the model does not write too much and thereby increase latency, we ask it to restrict its word count to 150 words.

```
You are a system that teaches ranking to other LLMs by giving an explanation. Your job is
     to take a number of passages and a query as input and output an ordered list, comma
    separated with the passage numbers inside () of the correct ranked order of these
    items. For example - (6,1,3,4,2,5)
The most relevant passage is provided in this array of 0s and 1s. This is a secret and
    provided only to you. Do not mention this array anywhere in your response. Remember
    to rank the passage that has a 1 in this array highest and justify the same - <ground
     truth>,
Below you will get the passages and query as input from the user. Using the hint of the
    array provided above, give strong reasoning in around 100-200 words of your ranking
    followed by the ranking itself. Remember that the LLM you are teaching to will not
    have access to the array of right ranking. It must learn from your explanation.
Lastly, give a short justification for the rankings of the other passages too. The
    important passage as seen in the secret array can be justified in around 50 words
    while the remaining 50 words can be used to justify the other passages' ranking.
DO NOT EXCEED 150 WORDS FOR YOUR RESPONSE AT ANY COST. ALSO DO NOT USE () ANYWHERE ELSE
    EXCEPT IN THE FINAL RANKING. YOUR RESPONSE MUST HAVE ONLY 1( AND 1) FOR THE FINAL
    LIST.
```

```

Given the following passages -
<Passage 1>
<Passage 2>
...

and the query -
<query>

Provide a ranking of the different passages in the following format (best_passage_number,
     second_best, ...).
```

The responses of this model were then concatenated with the original prompt given to GPT 3.5 and fine-tuned over 50 samples on the OpenAI fine-tuning platform. It was run for three epochs and the final training loss was 0.6503. This model was then used to perform ranking with the same prompt as before.

3.3 Evaluated Approaches

Within this research, the 4 algorithms that are mentioned in the previous sections have been evaluated. A short description of each algorithm is given below.

(1) **BM25:** Using an existing library[2], BM25 was first evaluated on the subset that was selected.
(2) **Embedding-Based Ranking:** The second model evaluated was implemented using OpenAI's Text Embedding Model (text-embedding-3-small) [3]. The first step involved converting both the queries and the answers into vector representations using the embedding model. Relevance ranking was then determined by calculating the cosine similarity between each query vector and the search result vectors. The higher the score, the higher the ranking.
(3) **Standard GPT-3.5 Ranking:** This approach used the large language model GPT-3.5 by directly giving it the queries and their corresponding search results as shown in the prompt above. It was then asked to rank the search results and return the correct list based on the passage numbers.
(4) **Fine-Tuned GPT-3.5 Ranking:** The last framework used GPT-3.5 by fine-tuning it with a test set of 50 random descriptive queries. GPT-4 was provided with this test set (and the corresponding labels) to explain why the answer labelled with 1 is the most relevant to the query. Using this test set and the explanation, GPT-3.5 was then fine-tuned with these test samples and the explanation by training.

3.4 Evaluation

The evaluation was done using the Mean Reciprocal Rank metric to measure how well each algorithm performs. This metric looks at the rank of the first relevant answer provided by each model for the selected subset. The MRR was selected since it was the most practical given the nature of our dataset that provided an array of 0s and 1s to indicate the selected document and not a ranking as such.

[2] https://pypi.org/project/rank-bm25/
[3] https://platform.openai.com/docs/guides/embeddings

Table 1. Summary Statistics of Dataset Columns

	Mean	Standard Deviation
BM25	0.38	0.30
GPT3.5	0.38	0.32
GPT3.5 Finetuned	0.41	0.35
Embeddings	0.46	0.37

4 EXPERIMENTAL RESULTS

The table consisting of the mean and standard deviation of the different approaches is provided in 1. Based on the mean value, the embedding model performs the best followed by the fine-tuned approach. However, using methods like mean and standard deviation for a metric like MRR can be misleading and thus further statistical tests must be conducted.

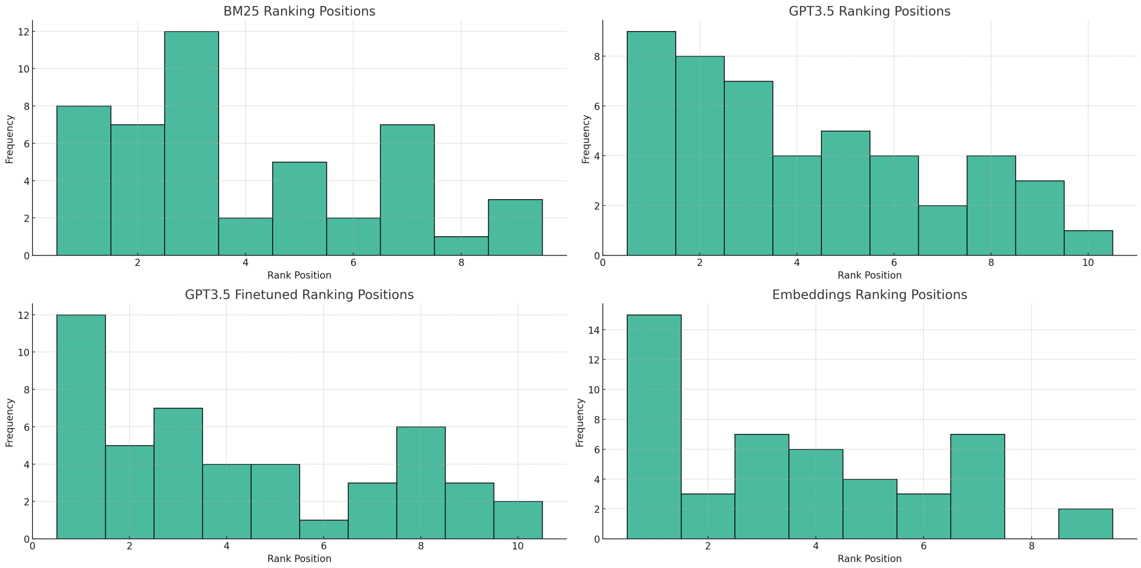

Fig. 2. The different rankings of the correct passage by each of the different approaches. It is worthwhile to note how GPT3.5 after fine-tuning tends to put more of the correct passages in the first column than the non-fine-tuned approach

The MRR scores for all 50 test samples were used to statistically analyse and reason out if there are any significant differences between the performances of the different rankings. We have one dependent variable - the MRR, one independent variable - the ranking approach, and 4 total ranking approaches (making it a categorical variable) all providing scores for the same query making it a within-subjects factor. Thus we first test for normality using the Shapiro-Wilk test for normality and if the data is normally distributed we do the paired t-test for all 5 pairs. Otherwise, we proceed with the Wilcoxon signed-rank Test.

The Shapiro-Wilk test of normality indicates that none of the 4 ranking approaches are normally distributed which makes sense looking at the calculation fo the MRR as 1/rank. The results of the same are as follows -

(1) GPT3.5 base : statistic=0.8122442960739136, pvalue=1.406954424965079e-06

(2) GPT3.5 Finetuned : statistic=0.7926263809204102, pvalue=4.886751412414014e-07
(3) BM25 : statistic=0.8091291785240173, pvalue=1.1845128256027238e-06
(4) Embeddings : statistic=0.7830525636672974, pvalue=2.9799912226735614e-07

Based on these results it is clear that we need to do the Wilcoxon signed-rank test. The results are presented in Table 2. As we can see, none of the differences are statistically significant.

Comparison	Statistic	P-value
GPT3.5 vs. BM25	408.0	0.979
GPT3.5 vs. Embeddings	306.0	0.241
GPT3.5 vs. GPT3.5 Finetuned	181.0	0.189
GPT3.5 Finetuned vs. BM25	345.5	0.717
GPT3.5 Finetuned vs. Embeddings	247.5	0.393

Table 2. Wilcoxon signed-rank test results for MRR comparisons.

Based on the results above, we note that our proposed approach of fine-tuning GPT3.5 with answers from GPT4 Turbo does not perform statistically better than any of the other ranking approaches. Figure 2 however gives some insights into the ranking of the different approaches on 50 samples of the test set. We observe how the fine-tuned approach tends to get better at putting the right query in the first position. Overall in terms of the average and median MRR however, the embedding model appears to rank better than the rest.

However, it is important to note the inherent limitations of our study. First, the number of descriptive queries used in this research is just a relatively small subset of 50 from the MS Marco dataset, and this could be a problem with query diversity, especially since only descriptive queries have been considered. This remarkably small sample of queries was primarily motivated by the costs associated with fine-tuning and running the GPT 3.5 models for inference and running GPT4 Turbo for creating the fine-tuning system. We recommend this study be performed once again with more training samples generated by GPT4 and tested on a larger and more diverse sample of queries and passages. Furthermore, we have currently utilised only the MS Marco dataset for our training and evaluation. Testing this approach on other datasets and for different applications could shed more light on the credibility and strength of fine-tuning to teach our model ranking using a more powerful LLM.

We also used only one metric for evaluating the performance of the models - and a rather controversial one at that. [3] explains why MRR is not a reliable metric to use and due to the very manner of its calculation is an ordinal scale rather than a continuous scale. Using other metrics such as nDCG or the metric of MFR (Mean First Relevant) as explained in [3] could be more promising and ensure we can mitigate the problems of using MRR for our evaluation criterion.

However, one problem which we believe will persist despite these changes is that of latency. LLMs, including smaller models, are still significantly slower than the state-of-the-art for most ranking applications. Even with a smaller and faster model, we believe the speed will still lag behind other ranking approaches - especially if Chain-Of-Thought reasoning is incorporated into the model. However, there may still be certain applications like chatbots or other retrieval systems where some delay is more permissible than others making them potential candidates for our approach.

Thus, to address these limitations and deepen our understanding of the different ranking algorithm's performance, we recommend a more holistic approach for any future research. This includes - a) using a wider range of queries and increasing the sample size of both the training and testing test, b) more comprehensive evaluation criteria and c) investigating different applications of rankings using diverse datasets to ascertain if our approach is better for specific use-cases. We believe that with these steps undertaken, our approach still holds promise and could potentially be used for specific applications. Our investigation lays the groundwork for future research into fine-tuning smaller models using larger LLMs for ranking.

5 CONCLUSION

This comparative study has shed light on the potential areas for growth and future research in the field of information retrieval. While the fine-tuned GPT-3.5 model did not outperform current ranking approaches, it did underscore large language models (LLMs) ability to process descriptive queries and grasp semantic understanding with reasonable accuracy. Further refinement in fine-tuning methodologies is necessary. Taking into consideration the limitations of our experiment, notably the reliance on a narrow selection of queries and a single dataset, this research might not fully capture the diversity and complexity of real-world information needs. Hence, some further extensions would include expanding the variety of queries examined, exploring a wider array of datasets, and increasing our sample sizes to boost the credibility and applicability of our findings. Delving deeper into model comparison and consistency across various scenarios also ensures that these technologies can be reliably deployed in enhancing the efficiency and relevance of search engines and information retrieval systems. In all, this study serves as an exploratory step into the transformative impacts these innovations in generative AI may hold for the future of ranking. We hope that this study will thus make information retrieval to be more efficient and effective.

6 SELF-ASSESSMENT OF CONTRIBUTIONS

- **Ahmed:** During this research, my primary contributions involved evaluating the deep learning algorithms and writing the introduction and related works sections of the report.
- **Chandran:** I came up with the idea of ranking based on fine-tuning LLMs, generated the fine-tuning table, as well as organizing and writing the methodology section.
- **Noel:** Primary contributions include applying and evaluating traditional IR techniques as well as writing abstract, experimental results and conclusion of the report.
- **Patricia:** Primary contributions include applying and evaluating traditional IR techniques as well as writing abstract, experimental results and conclusion of the report.

REFERENCES

[1] Emmanuele Chersoni, Enrico Santus, Chu-Ren Huang, and Alessandro Lenci. 2021. Decoding Word Embeddings with Brain-Based Semantic Features. *Computational Linguistics* 47 (2021), 1–36. https://doi.org/10.1162/coli_a_00412

[2] Kevin Duh and K. Kirchhoff. 2011. Semi-supervised ranking for document retrieval. *Comput. Speech Lang.* 25 (2011), 261–281. https://doi.org/10.1016/j.csl.2010.05.002

[3] Norbert Fuhr. 2018. Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum* 51, 3 (feb 2018), 32–41. https://doi.org/10.1145/3190580.3190586

[4] J. Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2019. A Deep Look into Neural Ranking Models for Information Retrieval. *ArXiv* abs/1903.06902 (2019). https://doi.org/10.1016/J.IPM.2019.102067

[5] P. Jain, R. Ross, and Bianca Schoen-Phelan. 2019. Estimating Distributed Representation Performance in Disaster-Related Social Media Classification. *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2019), 723–727. https://doi.org/10.1145/3341161.3343680

[6] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[7] Hang Li. 2011. Learning to Rank for Information Retrieval and Natural Language Processing. *Synthesis Lectures on Human Language Technologies* 4 (2011), 1–113. https://doi.org/10.1007/978-3-031-02141-1

[8] Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726* (2022).

[9] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (2009). https://doi.org/10.1007/978-3-642-14267-3

[10] José Antonio Hernández López, Carlos Durá, and Jesús Sánchez Cuadrado. 2023. Word Embeddings for Model-Driven Engineering. *2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS)* (2023), 151–161. https://doi.org/10.1109/MODELS58315.2023.00036

[11] Saurav Manchanda, Mohit Sharma, and G. Karypis. 2019. Intent Term Weighting in E-commerce Queries. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019). https://doi.org/10.1145/3357384.3358151

[12] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir R. Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual Generalization through Multitask Finetuning. (2023), 15991–16111. https://doi.org/10.18653/v1/2023.acl-long.891

[13] Hosein Rezaei. 2022. Word Embeddings Are Capable of Capturing Rhythmic Similarity of Words. *ArXiv* abs/2204.04833 (2022). https://doi.org/10.48550/arXiv.2204.04833

[14] S. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3 (2009), 333–389. https://doi.org/10.1561/1500000019

[15] Syandra Sari and M. Adriani. 2014. Learning to rank for determining relevant document in Indonesian-English cross language information retrieval using BM25. *2014 International Conference on Advanced Computer Science and Information System* (2014), 309–314. https://doi.org/10.1109/ICACSIS.2014.7065896

[16] Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Z. Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *ArXiv* abs/2304.09542 (2023). https://doi.org/10.48550/arXiv.2304.09542

[17] K. Svore and C. Burges. 2009. A machine learning approach for improved BM25 retrieval. *Proceedings of the 18th ACM conference on Information and knowledge management* (2009). https://doi.org/10.1145/1645953.1646237

[18] D. Tran and Alexandros Iosifidis. 2019. Learning to Rank: A Progressive Neural Network Learning Approach. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), 8355–8359. https://doi.org/10.1109/ICASSP.2019.8683711

[19] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* 29 (2017), 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[21] Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, E. Learned-Miller, and J. Kamps. 2018. From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (2018). https://doi.org/10.1145/3269206.3271800

[22] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji rong Wen. 2023. Large Language Models for Information Retrieval: A Survey. *ArXiv* abs/2308.07107 (2023). https://doi.org/10.48550/arXiv.2308.07107