UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Elétrica e de Computação

Fernando Ferraretto Silva

ExaRanker: Synthetic Explanations Improve Neural Rankers

ExaRanker: Explicações Sintéticas melhoram ranqueadores neurais

Campinas
2024

Fernando Ferraretto Silva

ExaRanker: Synthetic Explanations Improve Neural Rankers

ExaRanker: Explicações Sintéticas melhoram ranqueadores neurais

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Automação.

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering, in the field of Automation.

Supervisor/Orientador: Roberto de Alencar Lotufo
Co-supervisor/Coorientador: Rodrigo Frassetto Nogueira

ESTE TRABALHO CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO/TESE DEFENDIDA PELO ALUNO FERNANDO FERRARETTO SILVA, E ORIENTADA PELO PROF. DR. ROBERTO DE ALENCAR LOTUFO

Campinas
2024

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca da Área de Engenharia e Arquitetura
Elizangela Aparecida dos Santos Souza - CRB 8/8098

Si38e

Ferraretto Silva, Fernando, 1990-
ExaRanker : synthetic explanations improve neural rankers / Fernando Ferraretto Silva. – Campinas, SP : [s.n.], 2024.

Orientador: Roberto de Alencar Lotufo.
Coorientador: Rodrigo Frassetto Nogueira.
Dissertação (mestrado) – Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de linguagem natural. 2. Aprendizado de máquina. 3. Sistemas de armazenamento e recuperação de informações. I. Lotufo, Roberto de Alencar. II. Nogueira, Rodrigo Frassetto, 1986-. III. Universidade Estadual de Campinas (UNICAMP). Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações Complementares

Título em outro idioma: ExaRanker : explicações sintéticas melhoram ranqueadores neurais
Palavras-chave em inglês:
Natural language processing
Aprendizado de máquina
Sistemas de armazenamento e recuperação de informações
Área de concentração: Automação
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Rodrigo Frassetto Nogueira [Coorientador]
Razieh Negin Rahimi
Sandra Eliza Fontes de Avila
Data de defesa: 10-06-2024
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0009-0005-4674-8705
- Currículo Lattes do autor: http://lattes.cnpq.br/2198145128017900

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Fernando Ferraretto Silva **RA:** 081389

Data da defesa: 10/06/2024

Título da Dissertação: "ExaRanker: Explicações Sintéticas melhoram ranqueadores neurais”

Dissertation Title: “ExaRanker: Synthetic Explanations Improve Neural Rankers”

Prof. Dr. Rodrigo Frassetto Nogueira (Presidente, FEEC/UNICAMP)
Prof. Dra. Sandra Eliza Fontes de Avila (IC/UNICAMP)
Prof. Dra. Razieh Negin Rahimi (CICS/UMass)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Acknowledgements

I am deeply grateful for the journey I've undertaken in my studies since childhood. My father, Valter, and grandfather, Oswaldo, provided unwavering support and encouragement, fostering my curiosity and passion for learning. Their guidance was a significant source of strength from a rational perspective. On the other hand, my mother, Cida, grandmothers, Adamil and Selenia, and grandfather, Moyses, nurtured my intuition and motivation, inspiring me to persevere and achieve my goals.

Though the path has been long, I take great pride in my accomplishments and believe my family would be proud of me as well, wherever they may be. I wish to extend my heartfelt thanks to everyone around me—my uncles, aunts, cousins, and all who have stood by my side throughout this journey.

Academically, I am especially grateful to my advisors, Lotufo and Rodrigo, whose inspiration and support were invaluable. During moments when I nearly gave up, their encouragement motivated me to continue and transformed the experience into a pleasure journey.

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil.

O presente trabalho foi realizado com apoio do processo nº 2022/01640-2, da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

"Os circuitos de consagração social serão tanto mais eficazes, quanto maior a distância do objeto consagrado."

"The circuits of social consecration will be all the more effective, the greater the social distance from the consecrated object."

Pierre Bourdieu

Resumo

Esta tese de mestrado investiga a aplicação inovadora de modelos de linguagem de grande escala (LLMs) em contextos de recuperação de informações (IR). Estudos recentes destacaram a eficácia da alavancagem de LLMs para gerar explicações em linguagem natural antes de fornecer respostas, resultando em melhorias de desempenho em diversas tarefas. Baseando-se nessa premissa, este estudo explora o potencial impacto da integração de explicações em linguagem natural como dados suplementares para ranqueadores neurais.

Por meio de uma experimentação abrangente que engloba diversos modelos de linguagem e tamanhos de conjunto de dados, esta pesquisa examina detalhadamente a dinâmica da ampliação de dados no domínio da IR. Os resultados demonstram os benefícios consistentes e tangíveis da incorporação de explicações no processo de treinamento. Além disso, o estudo revela que à medida que a escala dos modelos de linguagem aumenta, também aumentam os ganhos de desempenho, destacando o papel crucial da escala na eficácia do modelo. No entanto, também reconhece as considerações críticas em torno dos tamanhos destes modelos, incluindo o consumo de tempo e as implicações de custos, que são explorados minuciosamente em diferentes cenários.

O método proposto para ampliação de dados apresenta uma solução alternativa ao enfrentar o desafio da escassez de dados, acelerando o processo de aprendizado e aplicando os benefícios da linguagem natural no treinamento de ranqueadores neurais. Ilustrados por resultados que ultrapassam as referências alvo em uma avaliação zero-shot, esses achados afirmam a eficácia da metodologia e defendem uma adoção mais ampla de técnicas de aumento de dados. Este estudo possibilita uma nova estratégia no campo da recuperação de informações, impulsionada pela integração de explicações em linguagem natural na utilização de ranqueadores neurais, além habilitar a aplicação deste método em diferentes problemas e contextos.

Palavras-chave: Processamento de Linguagem Natural; Transformers; Aprendizado de Máquina; Modelo de Linguagem de Grande Escala; Recuperação de Informações.

Abstract

This master's thesis delves into the innovative application of large language models (LLMs) within information retrieval (IR) contexts. Recent studies have underscored the effectiveness of leveraging LLMs to generate natural language explanations before outputting answers, leading to performance improvements across diverse reasoning tasks. Building upon this foundation, the study explores the potential impact of integrating natural language explanations as supplementary labels within neural rankers.

Through comprehensive experimentation encompassing diverse language models and dataset sizes, this research examines in details the intricate dynamics of data augmentation in the IR domain. The findings demonstrate the consistent and tangible benefits of incorporating explanations into the training process. Moreover, the study reveals that as the scale of language models expands, so too do the performance gains, highlighting the pivotal role of scale in model efficacy. However, it also acknowledges the critical considerations surrounding LLM sizes, including time consumption and cost implications, which are thoroughly explored in different scenarios.

The proposed data augmentation approach presents an alternative solution to address the challenge of data scarcity, accelerating the learning process by leveraging natural language to finetune language models. Illustrated by results surpassing the target baseline in a zero-shot evaluation, these findings affirm the efficacy of the methodology and advocate for broader adoption of data augmentation techniques. This thesis enables a new strategy in the field of information retrieval, driven by the seamless integration of natural language explanations into neural rankers and also enabling the application of this method in different problems and contexts.

Keywords: Natural Language Processing; Transformers; Machine learning; Large Language Model; Information Retrieval.

List of Figures

List of Tables

Contents

1 Introduction

As the amount of data, texts, websites, and journals continue to grow, Information Retrieval (IR) importance is increasing significantly. The data can come from diverse sources, ranging from scientific papers to personal social media accounts and devices connected in the internet. Recent researches estimates the so called datasphere, which is the amount of data created, captured, and replicated in any digital media across the world, will grown exponentially as show in Figure 1 from IDC (Internet Data Center) research[43].

This data expansion happens mainly due to the internet's popularity, world globalization and internet of things which are leading to an enormous amount of data being available on almost any topic. Despite the benefits of having vast amounts of information readily available, effectively filtering and selecting the relevant data poses a challenge.

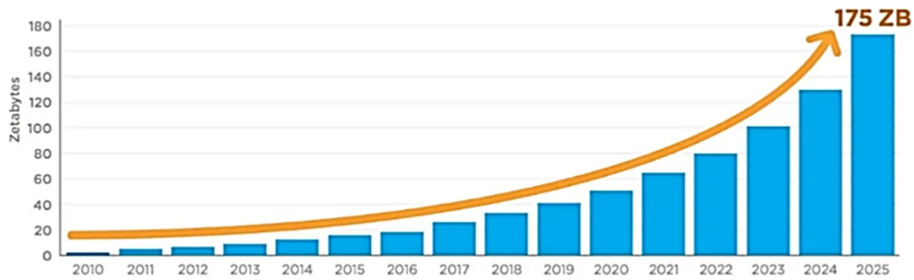

Figure 1: Data available in the digital platforms - from IDC Data Age 2025.

In such scenario, IR emerges as a crucial method to select pertinent information sources from a pool of potential candidates. Without a glossary or method to sort the data based on a desired topic, the abundance of information would be rendered useless since there would be no means to filter or select the desired content.

The information retrieval field has different application and solutions as briefly shown in Figure 2. From indexing to search formulation, IR has been introduced in the big data environment to select the most relevant information passages for a given query. IR is critical for the effective implementation of search engines and facilitates relevant information selection and usability. However, it is a complex task due to the diversity and amount of data being processed by the IR systems.

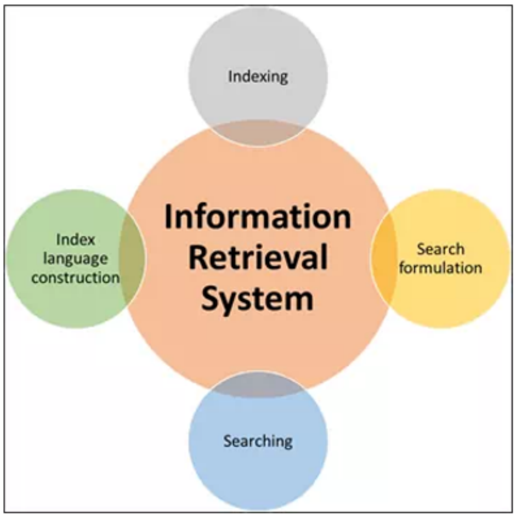

Figure 2: Information Retrieval main applications.

In this continuous growth and constantly changing work scenario, artificial intelligence and its various derivatives, such as Large Language Models (LLMs), present promising strategies for enhancing the effectiveness and applications of search engines. Neural models, in particular, have gained widespread usage in addressing complex problems within the field of information retrieval [30], where traditional syntactical and lexical approaches may fall short in diverse contexts and information structures. Recent advancements in Natural Language Processing (NLP) have further amplified the effectiveness of these neural models, with AI proving beneficial across a wide range of tasks. Notably, advances in self-supervised training and the introduction of the Transformer architecture [62] signify a remarkable leap forward in the capabilities and effectiveness of IR models.

Transformer models have proven to be highly effective in complex language processing tasks, including the field of information retrieval. The fundamental approach is to train these models on large datasets for specific tasks in order to improve their effectivenes. Referred to as neural rankers, these models are based in a neural network trained to learn suitable representations for inputs and ranking functions using neural networks, rather than relying solely on matching scores between queries and passages based on the presence of query terms within each passage. A base functionaly of neural rankers is illustrated in Figure 3. The neural network used varies between the different rankers and a range of deep learning models has been proposed [8, 9, 20], each presenting a distinct set of neural network components to extract features utilized for ranking.

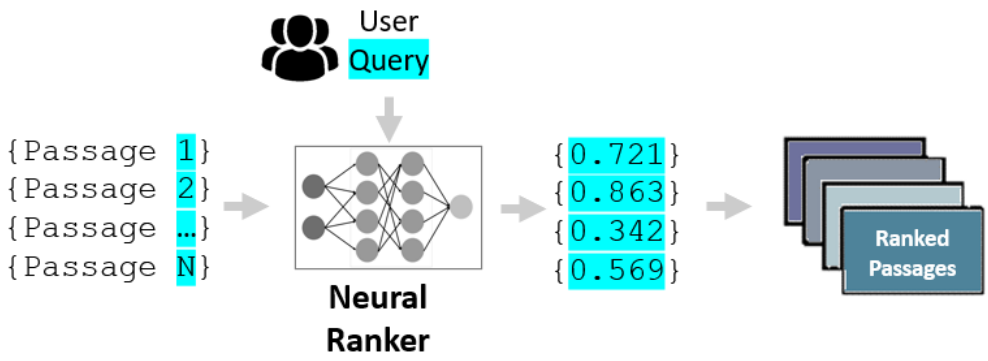

Figure 3: Neural ranker architecture. Given a query and N passages, the output score from a neural network is used to rank.

Specifically in IR experiments, pretrained Transformer such as BERT [11] and T5 [40] have been finetuned on hundreds of thousands of examples, leading to significant improvements [30, 37, 33, 28, 19, 66, 69, 23, 17, 32, 51, 22, 56, 72]. When queries and documents from a given task closely resemble those in the finetuning data, a model is likely to perform better than unsupervised models.

Nonetheless, deep learning methods for IR solutions depend heavily on the size of the available data used to finetune models, including Transformer based. Neural retrievers that are finetuned on large datasets outperform statistical models by a significant margin. For instance, a monoT5 [38] reranker trained with large quantities of labeled examples from MS MARCO [2] outperforms the statistical model BM25 [45] in 15 out of 18 datasets of the BEIR benchmark [48, 47].

However, when the number of labeled examples is limited, the effectiveness of the model decreases significantly. For example, a BERT reranker finetuned on a mere 10k query-relevant passage pairs performs only slightly better than BM25 on the MS MARCO passage ranking benchmark [38]. Increasing the size of the model [48] or pretraining it on IR-specific objectives [25, 18] can help to reduce the need for extensive finetuning data, but this comes at the cost of increased computational resources.

Likewise, finetuning neural rankers using only categorical labels, such as “true“ or “false“ is directly tied to the need for large amounts of training data. This is because the limited information supplied by these binary labels does not provide additional context or nuance to the learning process. To illustrate, it would be challenging to teach a person to evaluate the relevance of passages to queries using only the words “true“ or “false“ for each query-passage pair. Providing natural language explanations for why a passage is relevant or not to a given query would be a more efficient way to facilitate learning, rather than relying solely on binary classification and trial-and-error.

In this research study, we introduce a novel method for training retrieval models that employs natural language explanations as additional labels. This approach has several benefits, including reducing the need for extensive training examples. By utilizing natural language explanations, we aim to bring the language benefits observed in other NLP tasks to the IR domain.

It is well-known that NLP advancements have had a positive impact on a wide range of activities, and our proposed method is a step towards integrating these benefits into information retrieval. Furthermore, the usage of language inputs to augment categorical datasets may hold promise for various other tasks in future research.

Also, dataset size is often a limitation for applying AI solutions to a variety of different tasks. However, our proposed augmentation method can help to mitigate this limitation by enabling more tasks to be evaluated using neural models. Overall, we believe that our approach holds significant potential for improving retrieval effectivenes while reducing the need for large amounts of labeled data, which could have implications for a range of real-world applications.

To incorporate natural language, like explanations, as additional labels for training retrieval models, we need to generate this information as it is usually not available in the datasets. This can be a laborious and time-consuming process if done manually, which could be a potential obstacle for many researchers. To overcome this issue, an automated process is required for generating these explanations.

One way to generate these explanations is to use a few-shot LLM which can automatically augment the training examples with explanations. This approach allows IR practitioners to apply our method to other datasets without the need for manual annotation, thereby reducing the time and effort required to generate augmented data. By automating the generation of natural language explanations, we can enable the wider application of our method and promote the development of more effective retrieval models.

Integrating language models that produce text can elevate the quality of information retrieval tasks by generating explanations for ranked lists. However, this added capability might result in longer processing times, potentially slowing down the generation of categorical labels. Thus, it is vital to consider the trade-offs between the benefits of improved explanations and the potential impact on computational efficiency and performance.

Moreover, commercially LLMs, like GPT-3.5 [39], are often proprietary, potentially limiting methods due to the expenses involved in generating augmented datasets. Overcoming this limitation requires recognizing the advantages of utilizing natural language processing produced by open-source LLMs, which may be constrained by model size and consequently affect the quality of generated explanations. Despite potential lower quality in text generation, data augmentation remains beneficial for neural rankers, offering increased signal for improved comprehension and reranking of passages in information retrieval applications.

1.1 Main Contributions

The main contribution of this study is to introduce a novel method to augment categorical datasets with natural language explanations, thereby enhancing the effectiveness of retrieval models in an automated manner. The experimental results clearly demonstrate the advantages of incorporating language processing techniques for data augmentation in information retrieval research.

Notably, the study examines the relationship between data quantity and retrieval effectiveness, showcasing how language augmentation can address data scarcity issues and enhance model effectiveness. Additionally, a correlation is observed between the contributions of explanations and various LLM sizes. There exists a direct link between the enhancement in effectiveness and the quality of text automatically generated by these models for dataset augmentation.

While the method is flexible and applicable in various scenarios, it is crucial to acknowledge its limitations concerning the explanations generated. We do not claim that our method renders a retriever interpretable since we have not massively evaluated the accuracy of the generated explanations. Our objective is distinct from building interpretable retrievers; instead, we illustrate how explanations can enhance retrievers' effectiveness.

In summary, the main contributions of this novel method are listed below and explained in the next sections of this study,

1. A novel method for augmenting categorical datasets with natural language explanations, which can be used to improve information retrieval effectiveness. The method has been stressed over different data set sizes, prompts and models.

2. Large datasets generated from MSMARCO and augmented with explanations in regards of each query-passage relevance relation. It has been done using different LLMs and datasets sizes.

3. Improvements in neural rerankers effectivenes using a seq-to-seq strategy to finetune LLMs. The results surpass the strong baseline used as target for our study which was trained using categorical classification method.

4. Experiments to demonstrate the benefits of incorporating language processing techniques into information retrieval research. From small to large datasets, the experiments compare the neural ranker effectiveness with and without data augmented by text into categorical datasets.

5. The datasets augmented with explanation and the source code used to generate the explanations and apply this method has been public shared at `https://github.com/unicamp-dl/ExaRanker` for future studies.

6. Acceptance in the 46th International ACM SIGIR 2023 Conference on Research and Development in Information Retrieval [1] as a short paper [15]. Two additional [14, 16] papers related to this study have been published, aiming to disseminate our findings and allowing future works to leverage our augmented dataset for new researches and studies.

Overall, this method aims to enhance the models generalization capability and increase effectivenes across various IR datasets when evaluated in a zero-shot manner. It is also useful to speed up the learning process even when there is a shortage of data available, making it possible to tackle problems in a more efficient manner. Besides the contributions related to the IR field, the method can be extend in different areas and objectives beyond retrieving data.

2 Related Work

In recent studies, it has been found that augmenting language models with the capability to generate natural language rationales in a step-by-step manner can lead to a significant improvement in their effectivenes on a variety of reasoning tasks, either through LLMs optimization [42, 12, 27, 24] or exploring reasoning features [67, 71]. This involves the ability of the language model to explain its reasoning process in a clear and structured way, making it easier for humans to understand and evaluate its decision-making.

The results of these studies suggest that incorporating this feature into language models can result in more accurate and effectiveness outcomes in problem-solving. By providing a step-by-step explanation of its reasoning, a language model can provide greater transparency in its decision process, which is crucial for tasks that require a high degree of accuracy and reliability.

The addition of natural language rationales to language models represents a promising avenue for improving the capabilities and effectivenes of these models, and could have significant implications for a wide range of applications in fields such as healthcare, finance, and natural language understanding.

Despite the effectiveness of induced explanations in improving model effectivenes, it should be noted that the experiments conducted to evaluate their efficacy often involve models with billions of parameters. While these large models are indeed powerful, they may not always be practical for certain tasks, particularly those related to information retrieval.

For example, re-ranking 100 passages for a single query using a model with 175B parameters would require at least one minute on four A100 GPUs, which highlights the limitations of such models in terms of speed and practicality. Therefore, the computational resources and time required to run these models may be prohibitive for some applications, making them less accessible or feasible in certain contexts.

There is a need to strike a balance between model effectivenes and practicality when considering the use of induced explanations in real-world applications. More research is needed to explore alternative approaches that can achieve similar benefits with less computational overhead, making them more suitable for use in a wider range of scenarios and contexts.

In information retrieval tasks, the datasets often lack the specific language or terminology associated with the target label relevance. As a result, the potential benefits of language-based models may not be fully utilized. Despite this limitation, a significant amount of research has been devoted to developing techniques that can generate explanations to be integrated with ranked lists of results, it may be focused on interpretability techniques [53, 63, 54, 13, 73] or explanations for the ranked list [58, 49, 52, 64, 70]. These techniques provide a rationalization in the context of information retrieval, thereby improving the interpretability of the results.

For example, the GenEx model [41] generates noun phrases such as "impacts on Medicare tax" for a given query-document pair as briefly presented in Figure 4. Additionally, snippet generation can also be seen as a means of providing explanations for presenting certain results to the user [59, 61, 3, 6]. These methods enhance the overall effectivenes of information retrieval systems by improving the interpretability of the results, which is crucial for user satisfaction and trust. However it is limited to the result interpretability and not direct on the ranking effectiveness.

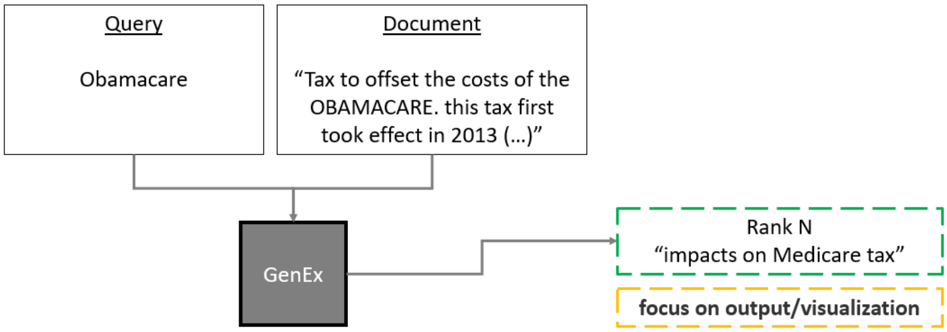

Figure 4: GenEx model - base functionality. This model generates output text to explain the relation between a query and a document.

By integrating explanation-generating techniques with ranked lists of results, information retrieval systems can provide users with more transparent and understandable results. This not only improves the overall user experience but also helps to bridge the gap between the language of the dataset and the language of the user, thereby facilitating more efficient and effective search queries.

While there have been several techniques developed to generate explanations for ranked lists of results in the context of information retrieval [46, 41], these techniques do not primarily focus on leveraging language models to support the finetuning process. Instead, their main goal is to produce qualitative language outputs that can enhance the interpretability and explainability of the results. However, generating high-quality language outputs can be a challenging task, especially when dealing with small datasets that lack reliable rationales. In such cases, the effectiveness of language-based models in supporting the finetuning process may be limited.

In our preliminary experiments, we observed that well-performing language models face difficulties in IR tasks where the target output lacks quality language signals. These models demonstrate efficient performance when processing text language, such as explanations, but may struggle when only a small text signal is expected, as in the case of categorical classification required by an IR problem.

Integrating language-based models into information retrieval tasks has the potential to enhance the quality of results by generating explanations for ranked lists of results. However, incorporating these models may also introduce additional processing time for generating explanations, which could potentially delay the production of categorical labels that are required for information retrieval tasks. Therefore, it is important to weigh the benefits of integrating language-based models for explanation generation against the possible drawbacks in terms of computational resources and effectivenes.

The focus of our work is to introduce a novel approach that utilizes LLMs to improve the quality of results in a ranking task without additional processing time in the reranker model. To achieve this, we draw upon existing techniques such as InPars [4, 26], Promptagator [10], and UPR [50]. However, our approach differs in that it leverages LLMs to enrich the target labels from training datasets with relevant information that pertains to the specific task at hand. Essentially, our approach aims to enhance the existing labels instead of generating new ones or scoring queries.

The InPars strategy uses a LLM to generated queries for a set of documents as illustrated in Figure 5. After this generation, a neural ranker is trained using the top queries based on the probabilistic score generated by the LLM. This method focus mainly in the dataset generation and represent an important and inspiring strategy to our work as they relies on the LLM output to finetune a neural ranker. However, InPars focus on improving the query quality and relevance using the benefits of large language models capacity to generate text.

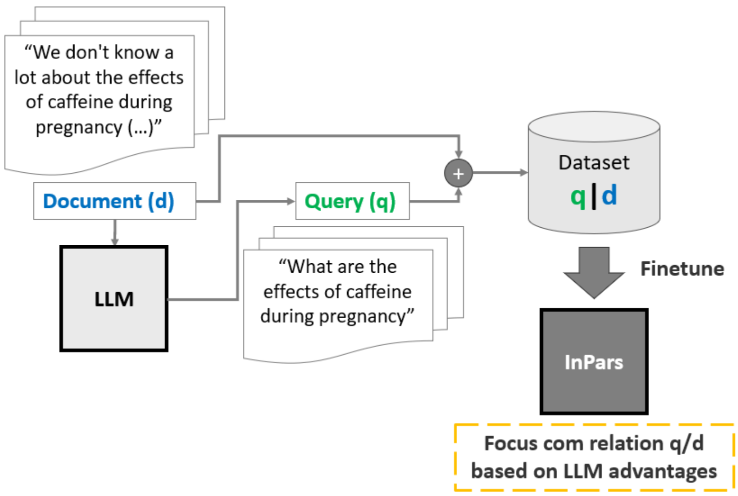

Figure 5: InPars model - method overview. The model uses synthetic queries from LLMs.

Similarly, the Promptagator focus on query generation by a LLM but using zero-shot and few-shot prompts. The queries generated are combined in a synthetic dataset and used to fine tune a neural ranker as presented in Figure 6. The method shows the benefits of combining queries using different strategies and generate by a LLM to improve the ranker effectiveness.

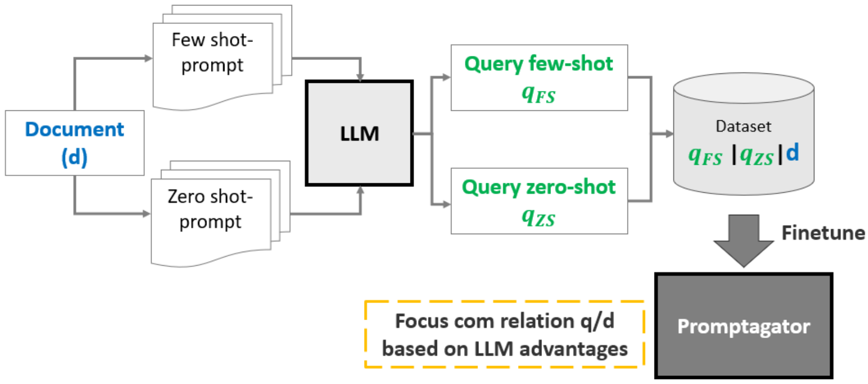

Figure 6: Promptagator model - method overview. This model uses different prompts to generate queries from LLMs.

On the other hand, UPR (Unsupervised Passage Re-ranker) use the LLM to generate queries for the list of passages being re-ranked. These synthetic queries are compared with the input query from the user and the passages re-ranked based on the similarity score (log likelihood) of the query generated by the LLM and the input query as shown in Figure 7. This method take the benefits of LLM text generation and compares the most-likely queries associated and generated by the model for each passage, using the LLM capacity to process natural language in a effective manner.

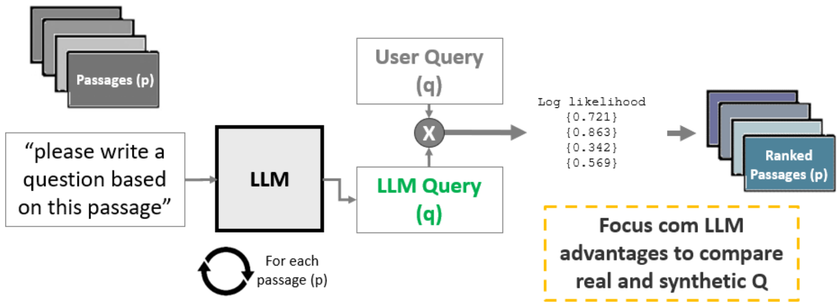

Figure 7: UPR method overview. The model compares user query and LLMs queries generated for all documents.

In summary, Figure 8 compares the previous works that inspired our method. Although these methods use LLM advantages, they are mainly focused in the query itself either to generate datasets or during the inference. Only GenEx tries to explain the results. Also, all methods are tackling the IR problem as a classification although generating text using the LLMs.

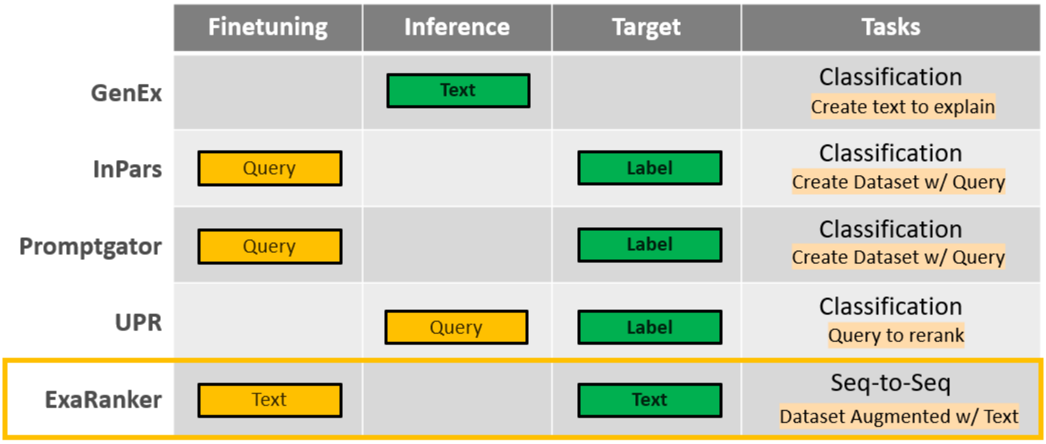

Figure 8: Comparison of related work purposes and methods. The last line shows ExaRanker which is a seq-to-seq model.

Our method, dubbed as ExaRanker, is a mix of these techniques, being inspired by the explanation in the results and dataset generation based on LLM outputs. By this combination, we move the problem from a classification task to a text generation problem. Using more text during the finetune is an opportunity to explore in a better way the benefits of NLP when using the LLMs in the IR field.

Another related technique is to use a graded relevance score [68, 35], such as assigning a continuous grade instead of a binary classification. However, in preliminary experiments, we have observed that LLM models are more efficient in generating natural language text rather than grading the relevance score between a query and a passage. Although we recognize the potential value of these existing methods, we believe that our approach offers a distinct and effective way to enhance ranking results. Moreover, we anticipate that future research may involve combining these strategies to achieve even better effectivenes in ranking tasks.

Finally, our approach uses naturally-occurring co-citations in scientific corpora to enhance document similarity models [36]. While our method may aid users in understanding a ranked list of results better, our primary focus is on improving the retrieval model's effectiveness, rather than creating an interpretable retriever. The accuracy and correctness of the generated explanations have not been evaluated, and hence, we cannot claim that our method renders a retriever interpretable.

3 Methodology

The proposed method is briefly outlined and illustrated in Figure 9. The approach begins by utilizing an LLM to augment the desired dataset with explanations related to its categorical label, which is either true or false. As already available in most information retrieval benchmarks, the method requires a baseline dataset that is composed by pairs of query-passage and its corresponding label, which is either relevant or not relevant.

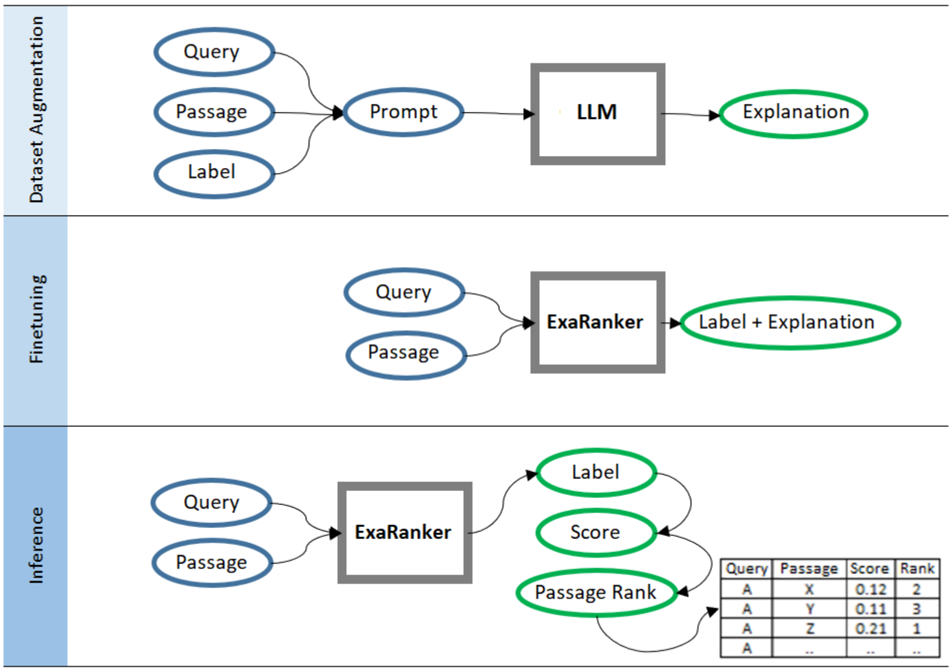

Figure 9: Method overview - 3 stages summary.

Once the baseline dataset has been selected, the next step is to choose a suitable LLM and an input prompt to bias the model to generate explanations for every query-passage sample given its label. The LLM is then used to generate explanations and augment the original dataset target. These augmented samples now form an augmented dataset which can be used to finetune a ranking model.

The proposed method aims to enhance the ranking model's by leveraging LLMs to generate high-quality explanations that are relevant to the specific IR task at hand. This approach is different from other methods that solely focus on providing users with an interpretable list of results. While the proposed method may help users better understand a ranked list of results, the primary objective is to improve the ranking model's effectiveness.

These explanations could be manually generated through human annotations but it would be cost and time prohibitive in datasets with thousands of samples. Using a LLM for this purpose, the process is entirely automated as the model can use its capacity to process natural language and generate the explanations efficiently, thereby saving time and cost.

In order to effectively utilize LLMs for generating explanations, it is important to refine the input prompt to convey to the model what constitutes an explanation and how it is expected to be generated. To this end, we propose the prompt input depicted in Figure 10. After conducting several trials, we formulated an input prompt that includes clear instructions and seven different examples consisting of a query, passage, answer, and explanation, all structured in a way that can be easily interpreted by the LLM.

Prompt - text

Instruction: explain if the passage is relevant to the question.
###
Example 1:
Question: What is a thyroid nodule
Passage: 1 Thyroid cancer: An uncommon form of cancer, thyroid cancer is usually curable. 2 Surgery, radiation, and hormone treatments may be used to treat thyroid cancer. 3 Thyroid nodule: A small abnormal mass or lump in the thyroid gland.
Final Answer: The passage is relevant to the question
Explanation: The question is what is a thyroid nodule. The last sentence of the passage describes the thyroid nodule.
...
###
Example 8:
Question: **{query}**
Passage: **{passage}**
Final Answer: The passage is **{"not " if label==False else ""}**relevant to the question
Explanation: **{explanation}**

Figure 10: Prompt used to generate explanations for a query-passage-label triple (presented in Python's f-string notation).

The eighth example, denoted as number 8, includes the query, passage, and label from the baseline dataset, and the model generates an explanation for the provided answer. By using this refined input prompt, we aim to optimize the LLM's ability to generate high-quality explanations that to be used in the downstream task.

The procedure of selecting the dataset samples, composing the prompt, sending it to the LLM, and extracting the generated explanation will be performed iteratively for each sample in the dataset. This process will continue until the entire dataset is processed, resulting in an augmented dataset that includes explanations for each sample. This method step is called as dataset augmentation and illustrated in Figure 11.

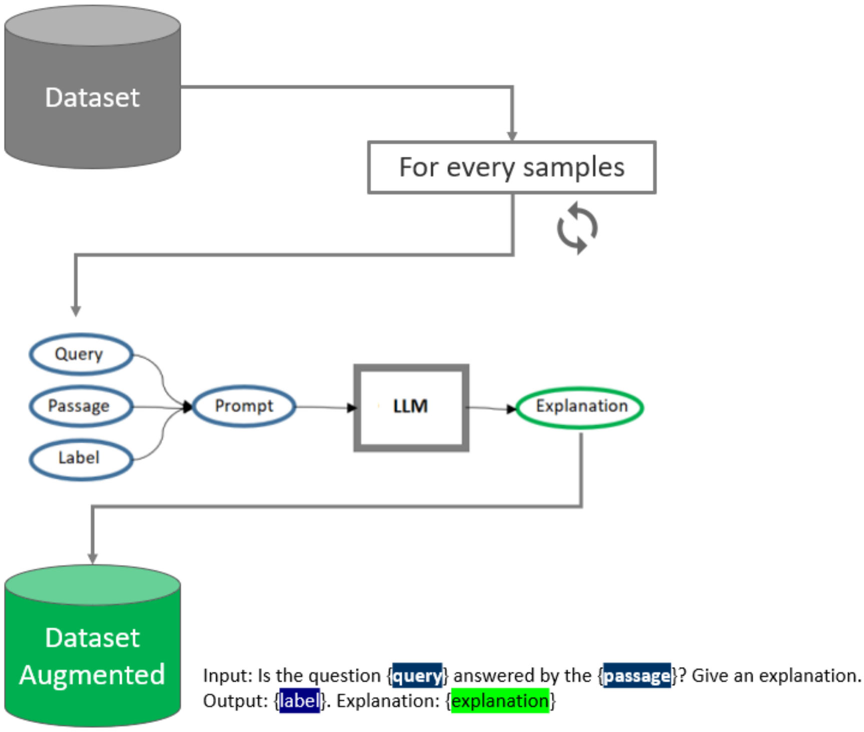

Figure 11: Dataset augmentation detailed.

It is crucial to highlight that our prompt explicitly provides the expected answer (relevant or not) to the LLM for each pair being evaluated. This approach is designed to direct the model's attention to the explanation instead of the answer. By doing so, we avoid three issues that may arise during the augmentation process.

Firstly, providing the answer in the prompt reduces the likelihood of LLMs incorrectly predict the answer, which could result in inaccurate explanations. Secondly, LLMs may generate weak text explanations if they are unclear about how to classify the presented pair. This can occur if the model's token scores for the "final answer" are low.

By providing the expected answer in the prompt, we help the model to better understand the relationship between the query and the passage, thereby producing stronger explanations. Figure 12 illustrate the scenario. The LLM is capable to generated explanations for the same query-passage pair considering it relevant or not relevant. However, in the case it is not a correct classification, i.e., different from the target, the explanation generated is weak, being ambiguous and not clear.

A similar situation is observed when the model has to decided the proper classification and then explain it. Comparing the third output presented in Figure 12 where the LLM did not received the final answer classification against the second output, where it was previously informed about the non-relevant relation, we see the second output with more accurate information regarding the query and the passage being assessed.

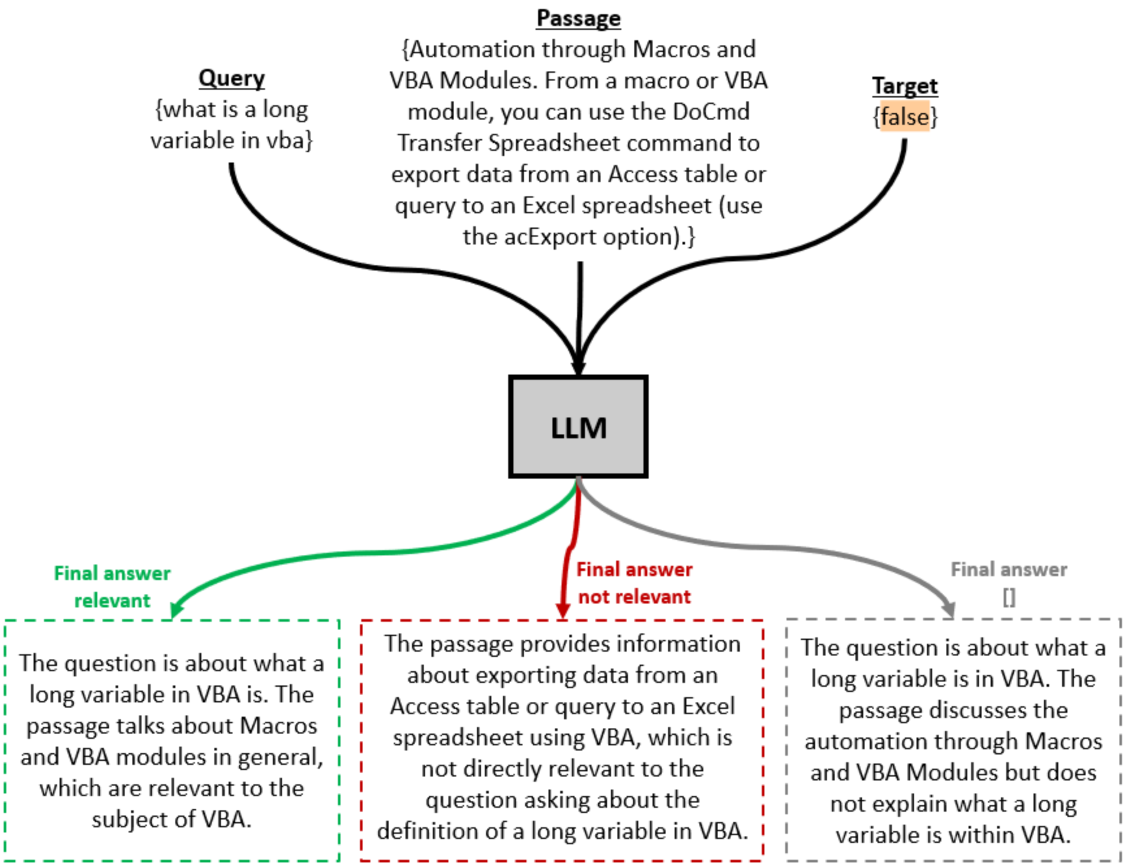

Figure 12: LLM explanations with correct, incorrect and without "Final Answer" in the prompt.

Another important technique used in the prompt is to provide to the models a text sequence as the final answer, i.e. ``The passage is (or not) relevant to the question.``, instead of a simple true/false label. By doing this, the models have more tokens related to the final answer in a language context. This is important because the models are designed to process language text and not just binary classifications. Additionally, by providing the models with more information, we can reduce the chances of the models mistaking the answer and generating a weak text explanation. Therefore, using plain text as the final answer is an important setup to improve the overall quality of the text output.

Once the explanations are generated, the baseline dataset will be augmented with the associated explanation for each query-passage pair, in addition to its binary label. This results in a new dataset where each sample contains a query, a passage, and a corresponding text explanation, in addition to its original label. As a consequence, the finetuning task of the LLM model changes from a binary classification task, where only the label is provided as the target, to a sequence-to-sequence task, where the target is a text explanation. This change in the target format is a crucial modification proposed in this study, as it allows the LLM model to learn and use the high-quality text explanations during the finetuning process.

As result of augmenting the baseline dataset with generated explanations, the proposed method provides more information to the model, which in turn accelerates the learning process.

The additional tokens in the augmented dataset transform the training task from a simple classification to a more complex sequence-to-sequence problem, thereby providing the model with more context and signals to learn from. As demonstrated in a variety of reasoning tasks, incorporating explanations into the model's output generation can significantly improve its effectivenes.

This approach moves the problem of information retrieval towards the sequence-to-sequence spectrum, where the models have previously been finetuned and optimized. Figure 13 illustrates the differences between the traditional neural rankers finetuning methods versus the ExaRanker. Our method enables the target model to be finetuned to produce the entire sequence and leverage language processing to enhance the assessment of correlation between each query-passage pair in the dataset. This strategy is expected to bring more benefits of LLM capability to process text instead of being used as simple classifier for a binary target.

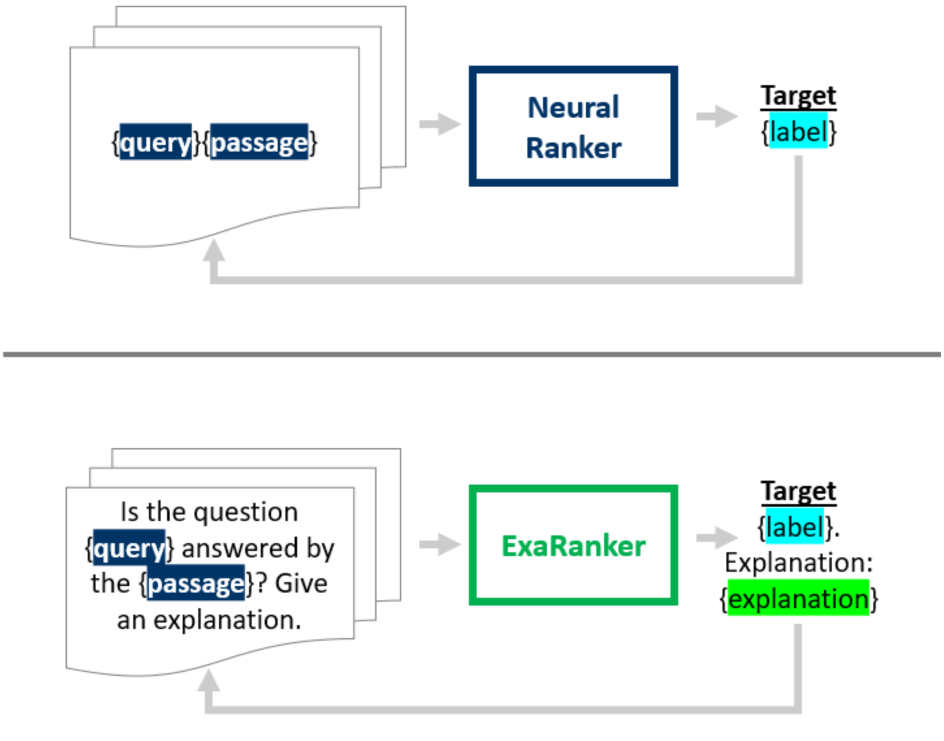

Figure 13: Finetune methods comparison: Neural Rankers vs ExaRanker. It highlights the different strategy in regard to the target.

Figure 14 shows the details of the finetuning process. It is a sequence-to-sequence model finetuned using the input/output templates specified below, as demonstrated using Python's f-string notation. The terms `{query}` and `{passage}` are the query-passage pair extracted from the baseline dataset. The `{label}` is `true` if the passage is relevant to the query and `false` otherwise. Finally, `{explanation}` is the one generated by the LLM as explained above.

Input: `Is the question {query} answered by the {passage}? Give an explanation.`

Output: `{label}. Explanation: {explanation}`

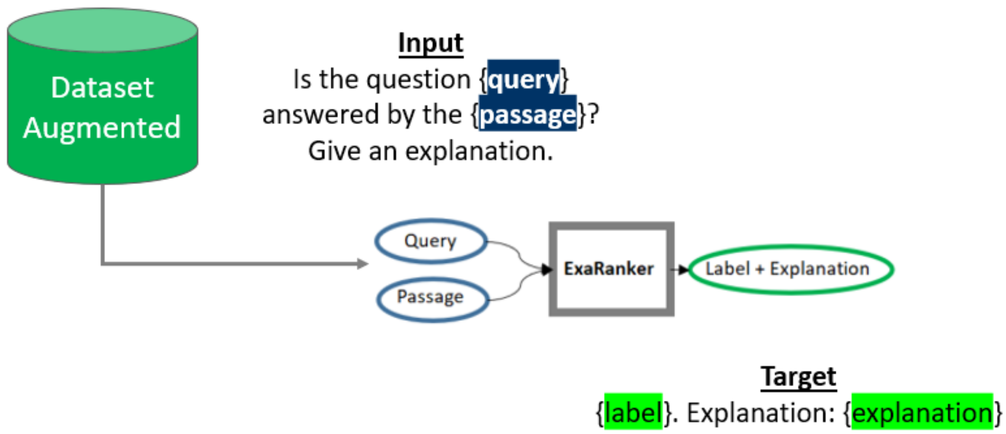

Figure 14: ExaRanker finetuning method detailed.

Since the model is being trained through text generation, the risk of overfitting is greatly reduced even when working with small datasets. The finetuned model is expected to be dynamic as it learns the classification task rationale and not just adjusts its weights to classify the provided dataset. Overfitting is a common problem when working with small categorical datasets and large models as it can easily adapt the weights to memorize each training sequence's classification, leading to failure when presented with an unknown sample. Although our method still uses a categorical dataset, it is now augmented with text and a sequence output.

In the inference phase, the finetuned model can be used to evaluate and rank a set of documents based on their relevance to a given query. While the finetuned model is capable of generating explanations, only the first token, i.e., the label token, is required. As demonstrated in Figure 15, our method can use only one token to reranking the documents which does not add any additional cost or time where compared to the traditional neural rerankers methods.

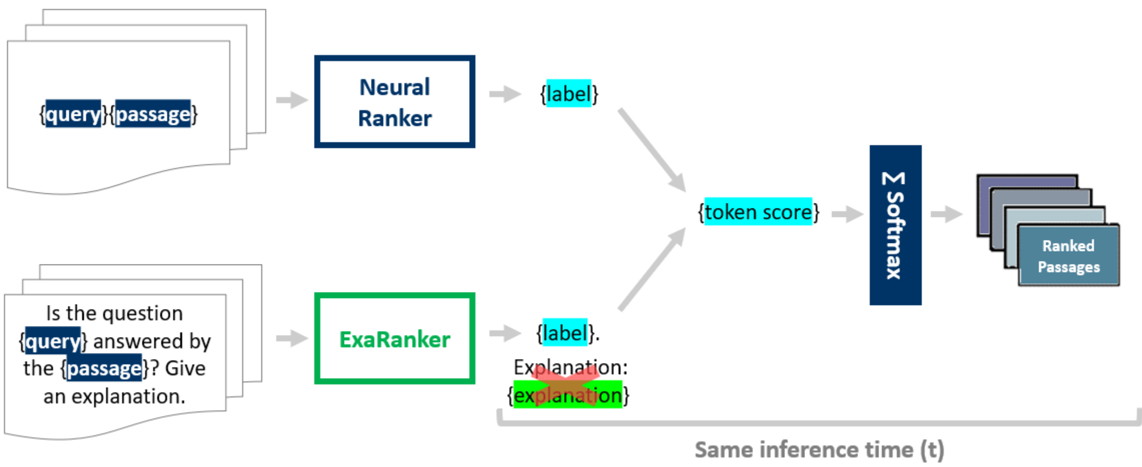

Figure 15: Inference time comparison: Neural Rankers vs ExaRanker in regard to inference time.

The model can be evaluated on the desired passage ranking task using the same input as designed before, and, as a consequence, it generates the same output pattern[1]. The inference strategy is detailed in Figure 16.

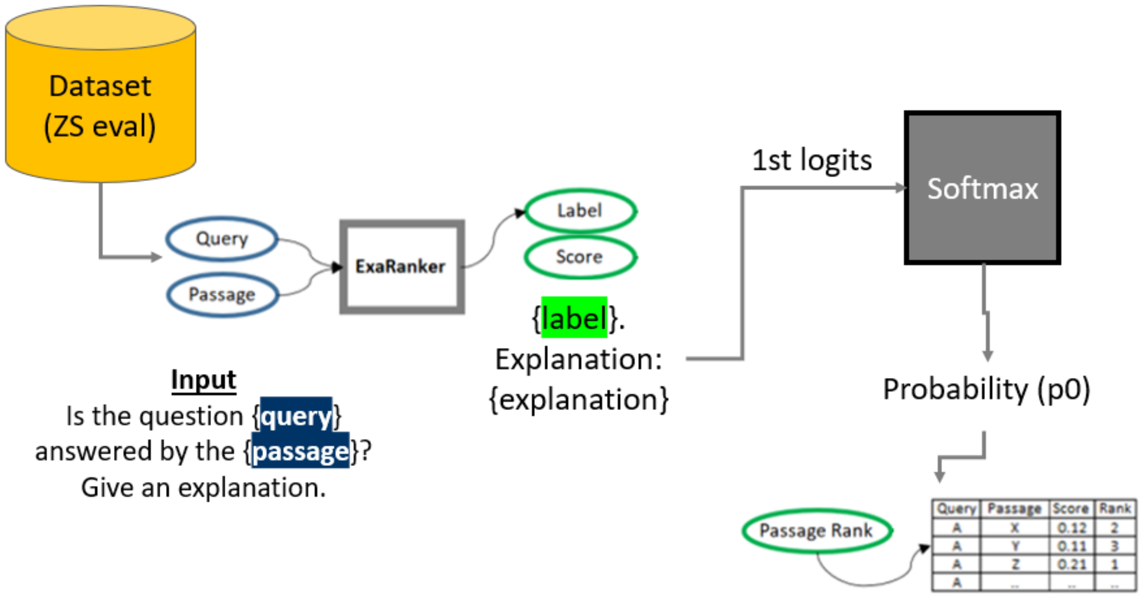

Figure 16: ExaRanker inference strategy detailed including the score calculation to rank the passages.

As presented, the probability of the first output token is used as the relevance score s for the query-passage pair which is calculated as follows. The unitary offset is used to set the passages with low probabilities of `true` or `false` classification with a score close to 1. However, the passages with high probability will be close to score 2 when it is `true`, and score 0 when its `false`.

$$s = \begin{cases} 1 + p_0, & \text{if } t_0 = \text{true} \\ 1 - p_0, & \text{if } t_0 = \text{false} \\ 0, & \text{otherwise} \end{cases}$$

where t_0 is the token generated in the first decoding step and p_0 is the probability assigned by the model to that token, i.e., the probability from the softmax after the logits.

The effectiveness of the reranker can be measured in different ways, but in general, the arrangement of a list of documents in the correct order of relevance is the objective, placing the most relevant documents at the top of the list and the least relevant ones at the bottom. This means that a good reranker should be able to distinguish between highly relevant and less relevant documents and order them accordingly.

[1]In our experiments the model always generates an output that matches the target template.

4 Experimental Setup

As previously presented in Figure 9, the experiment is divided into 3 stages: Dataset augmentation, finetuning and Inference. All details and parameters used in each stage are described in this section and visual presented for each phase in Figure 17.

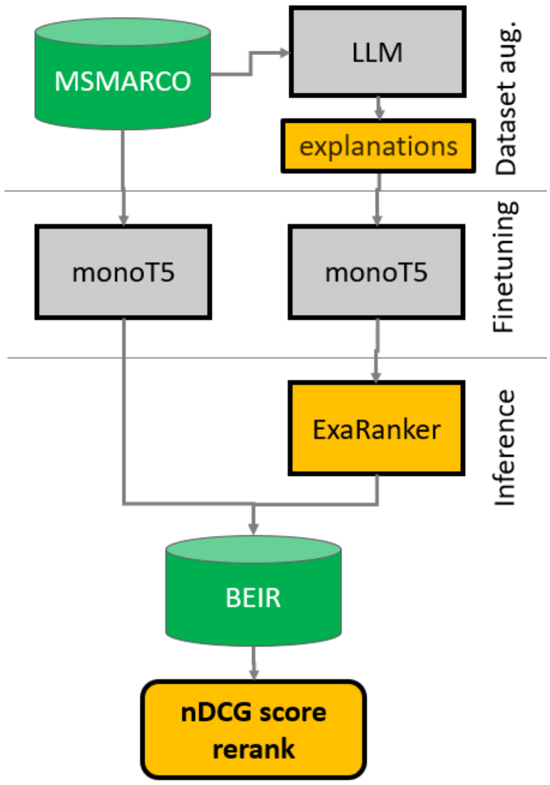

Figure 17: Experimental setup - ExaRanker and monoT5 (comparison model).

4.1 Dataset Augmentation

To begin our experiment, we first augment our dataset by selecting 300,000 query-relevant passage pairs and 300,000 query-non-relevant passage pairs at random from the training set of the MS MARCO passage ranking dataset. In order to generate explanations for these 600k pairs, we utilize the GPT-3.5 `text-davinci-002`[2], which allows us to infer explanations using the few-shot prompt described in Section 3.

Given the large volume of text involved, manually generating these explanations would be prohibitively time-consuming and costly. We employ a greedy decoding approach and limit the output to a maximum of 256 tokens. The few-shot prompt used in our experiment consists of 7 examples that were carefully selected from the MS MARCO training dataset. On average, this prompt has a length of 1400 tokens, which includes the 256 tokens needed for generating the explanation.

It is worth noting that as of March 2023, generating an explanation for each query-passage-

[2]beta.openai.com

label triple using the OpenAI API costs 0.028 USD, which total amounts to 16,800 USD. Once the explanations were generated, we augmented the original dataset by adding the explanation after the categorical label for each sample, thus creating the proposed input/output template.

Due to the high cost associate of using commercial LLMs to generate explanations, another 2 datasets were generated using open-source models. For these experiments, a smaller dataset consisting of 15k query-relevant and 15k query-non-relevant passage pairs from the same samples randomly chosen previously have been used.

After that, two open-source LLMs have been used for generating explanations and augmenting the dataset: llama-2-7B-chat-hf and llama-2-70b-chat-hf [60], using the same method and prompt as done with the GPT-3.5. In addition to the dataset with 30k samples, we created a larger version containing 100k samples. This dataset is composed of 50k pairs of query-passage relevant and another 50k non-relevant pairs. It is important to note that these models have not been previously finetuned on the IR task being evaluated in this study.

In total, five versions of augmented datasets with explanations were generated for this experiment, as detailed in Table 1.

LLM model	**Relevant Samples**	**Total Samples**
GPT-3.5	300k	600k
Llama-2-7B	15k	30k
Llama-2-7B	50k	100k
Llama-2-70B	15k	30k
Llama-2-70B	50k	100k

Table 1: Augmented datasets generated with explanations.

4.2 finetuning

The subsequent stage in our experiment involves finetuning a monoT5-base model, which was used as the initial starting point of our study. Although any sequence-to-sequence model could be used, we have selected monoT5 because it is currently close to the state-of-the-art in various NLP-related tasks.

During the finetuning process, we used the AdamW optimizer [31] for 30 epochs with a learning rate set of $3e-5$, weight decay of 0.01, and a batch size of 128 examples, which consisted 64 positives and 64 negatives. The maximum number of input and output tokens were each restricted to 512. During both training and inference, any sequences exceeding these limits were truncated. Figure 18 provides examples of input and output generated by the ExaRanker model after finetuning. Each model has been finetuned over different attempts, at least three different times, in order to have a more realistic scenario, mitigating the risk of optimal finetuned models in a random manner.

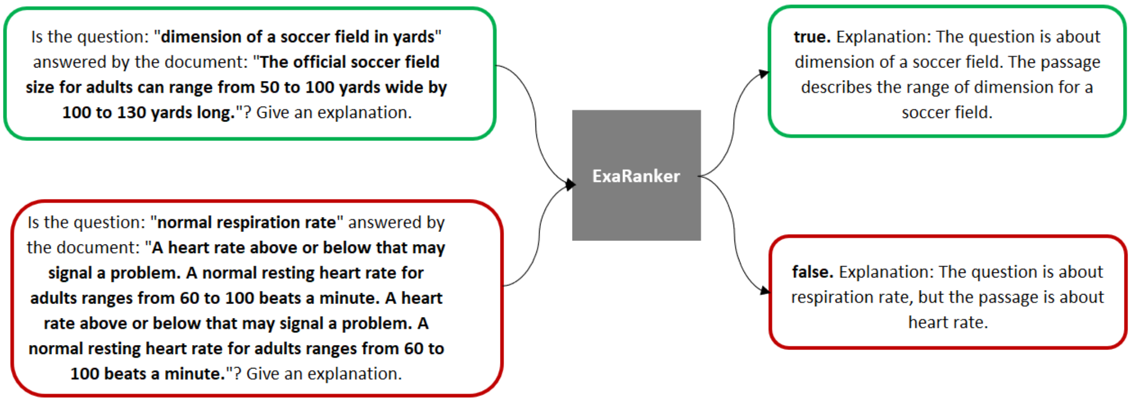

Figure 18: Illustration of input and generated outputs of a relevant (green) and non-relevant (red) query-passage pair.

To provide a basis of comparison, we also finetuned a monoT5-base model using the same dataset and hyperparameters as ExaRanker, but without incorporating any explanations into the target text. The input/output template used for this model was slightly modified, as the phrase "Give an explanation." was removed from the input, and the `explanation` tag was removed from the output. The resulting model will be referred to as the baseline model, and its effectiveness will be compared against ExaRanker models in Section 5. Figure 17 presented before provides a better visualization of the experimental setup, side by side, comparing the monoT5-base model and the ExaRanker.

4.3 Inference

To evaluate the effectiveness of our reranker, we adopted a widely used metric called the Normalized Discounted Cumulative Gain (nDCG) score. The nDCG score measures the quality of the reranked list by taking into account both the relevance of the documents and their position in the list. In the results presented in the following sections, we measured the quality of the reranked list using the nDCG score, this allowed us to assess the effectiveness of our method in improving the relevance ranking of documents in IR tasks.

The nDCG is calculated based on the Discounted Cumulative Gain (DCG) metric, which takes into account both the relevance of the recommended documents and their position in the list. Specifically, the DCG for a query "q" and a ranked list "R" of documents "d" is calculated as follows:

$$DCG(R,q) = \sum \frac{2^{rel(q,d)} - 1}{log_2(i+1)}$$

Then the nDCG is calculated dividing the DCG by the ideal DCG, which is the DCG for the perfect recommendation order for the documents lists.

$$nDCG(R,q) = \frac{DCG(R,q)}{IDCG(R,q)},$$

where IDCG represents the DCG for a perfect ordered list.

The nDCG value ranges from 0 to 1, with 1 indicating the perfect recommendation order. The metric can be calculated for different sets of documents, such as the first top 10 documents or the first 1000 documents. In our method evaluation, we focus on the nDCG@10, which evaluates the recommendation order for the first 10 documents.

In order to evaluate the benefits of explanations in a realistic scenario where training data is not available, we conduct a zero-shot evaluation on six datasets from the BEIR benchmark [57], namely Robust04 [65], TREC-COVID [44], DBPedia [21], FiQA [34], TREC-NEWS [55], and NFCorpus [5]. We have used TREC-DL 2020 [7] as a validation set to select the best checkpoint for the model evaluation in the different experiments. As described before, at least three different attempts were made for each experiment, with this study reporting the average results obtained. This approach mitigate the risk of reporting unrealistic results due to suboptimal performance during the finetuning phase.

We calculate the relevance score s for each query-passage pair based on the probability score generated by the model in the label token (the first token from the output). Using this relevance score, we order the passages and calculate the nDCG@10 score for the ranked list as the effectivenes metric evaluated in all datasets.

During the inference phase, we have set a limit on the output of the ExaRanker model. Only the label for the query-passage pair (i.e., the first token) is generated, and the full explanation text is omitted to save processing time. This is because the model was trained with a causal mask, meaning that only the tokens generated so far influence the prediction of the next token.

Therefore, the relevance scores calculated would be the same, regardless of whether the model generated an explanation. It means ExaRanker and the monoT5-base model have the same processing time, however, it is still possible to generate explanations from ExaRanker by decoding more tokens until the termination token (e.g., `<EOS>`) is generated. We have not evaluated the quality of the explanation generated as our focus was on improving the effectiveness of the retrieval model rather than building an interpretable retriever.

5 Results

The main results are presented in Table 2. As a baseline for comparison, we provide the results of the BM25 algorithm in the first two rows of the table. These results are obtained using the Pyserini toolkit [29] where the “flat” index concatenates the document titles and contents into a single field, while the “multifield” index stores them as separate fields. The BM25 algorithm is a statistical retrieval function that uses lexical matching of words between two texts to provide a similarity score. It can be used to rank a set of documents given a query using the score calculated by the BM25. This baseline does not employ any neural model and is purely based on statistical methods.

In the third row, we present a monoT5-base model that was finetuned for one epoch on 400k positive query-passage pairs from the MS MARCO dataset.[3] This model is used as a baseline for comparison with the rerankers reported in this work, which rerank 1000 documents retrieved from BM25’s “flat“ indexes.

Our analysis focuses on the number of positive query-passage pairs that require manual annotation, as this is a labor-intensive task when developing a search engine. It involves “experts“ in the domain to read and evaluate each query-passage pair to determine if the passage is relevant to the query. In contrast, negative query-passage pairs can be easily obtained through automatic selection using a retriever once the queries have been collected. This means that we do not need to have experts evaluate them manually. However, even for negative pairs, we still need to generate explanations in our method. The cost associated with generating explanations for negative pairs is likely to decline in the future as open-source LLMs become more widely available.

The fourth row in our results table showcases the outcomes obtained after finetuning the monoT5-base model using a selected dataset of 600,000 samples. Out of these, 300,000 are relevant pairs of query-passage that serve as our metric for measuring dataset size and cost. The fifth row displays the results after finetuning ExaRanker, a monoT5-base model, on the same dataset using explanations as augmentation. The hyperparameters used to train both models were kept the same, and we selected the best checkpoint as explained in Section 4 and reporting the average results of the attempts done.

The evaluation results demonstrate that the `ExaRanker` model surpasses the model without explanations in almost all datasets. The zero-shot evaluation shows an average improvement of 1.0 nDCG@10 points when compared to the model without explanations. The finetuned models exhibit much higher nDCG@10 scores in comparison to the BM25 baseline.

Additionally, the `ExaRanker` model outperforms the monoT5-400k model by an average of 0.6 nDCG@10 points. This baseline is the monoT5-base model, which was trained over the entire MS MARCO dataset, consisting of 400k relevant pairs. In contrast, our model is trained on a dataset with 300k relevant pairs but still achieves superior performance.

[3] `https://huggingface.co/castorini/monot5-base-msmarco-10k`

Table 2: Results (nDCG@10) and average zero-shot (all except DL 20). The column "Ft Pos." is the number of positive training examples on which the model was finetuned. In BM25 multifield, document titles are separated from content.

Model	Ft Pos.	DL 20	Robust	Covid	Dbp	FiQA	News	NFC	Avg ZS
BM25 flat	-	0.478	0.407	0.594	0.318	0.236	0.395	0.321	0.379
BM25 multifield	-	-	0.407	0.656	0.313	0.236	0.398	0.325	0.389
monoT5	400k	0.652	0.536	0.777	0.419	0.413	0.447	0.357	0.491
`monoT5`	300k	0.662	0.532	0.780	0.412	0.403	0.446	0.350	0.487
`ExaRanker`	300k	0.682	0.558	0.784	0.427	0.416	0.451	0.349	0.497

The results of our study clearly demonstrate the advantages of using explanations as a source of additional data during the training phase. We observed that by incorporating explanations, the effectiveness of the `ExaRanker` model improved significantly across all 7 datasets. This improvement is reflected in the nDCG@10 scores, which increased in the zero-shot evaluation.

Therefore, we can conclude that the use of explanations as a form of data augmentation is highly effective in providing more signal to the model during the finetuning process, making it more efficient in terms of the amount of data required to achieve high effectiveness. This is a promising result that can have important implications for the development of more effective and efficient search engines.

As a means to gain a deeper understanding of the benefits of using explanations during the finetuning process, we conducted a series of experiments with smaller datasets of 150k, 100k, 50k, 15k, 10k, 5k and 2.5k relevant pairs of query-passage. In each of these experiments, the datasets contained an equal number of positive and negative query-passage pairs, resulting in the smallest dataset comprising 5k samples and the largest reaching 300k. Our results presented on Table 3, show that the `ExaRanker` model consistently outperforms the model that was finetuned without explanations.

Specifically, when finetuning on a dataset of 150k positive examples, the `ExaRanker` model performed 1.1 points better than the monoT5 model. This improvement increased to 1.4 points when finetuning on 50k examples and to 2.6 points when finetuning on only 5k positive examples. These results clearly demonstrate the efficacy of using explanations to provide additional signal and improve the efficiency during the finetuning process. The use of explanations allows the model to succeed with much less training data.

Table 3: Results (nDCG@10) and average zero-shot (all except DL 20) using different datasets size. The column “Ft Pos.” is the number of positive training examples on which the model was finetuned. In BM25 multifield, document titles are separated from content.

Model	**Ft Pos.**	**DL 20**	**Robust**	**Covid**	**Dbp**	**FiQA**	**News**	**NFC**	**Avg ZS**
`monoT5`	150k	0.663	0.537	0.790	0.395	0.396	0.443	0.349	0.485
`ExaRanker`	150k	0.684	0.559	0.781	0.426	0.413	0.447	0.348	0.496
`monoT5`	100k	0.658	0.528	0.774	0.400	0.396	0.434	0.350	0.480
`ExaRanker`	100k	0.677	0.552	0.776	0.419	0.412	0.440	0.350	0.492
`monoT5`	50k	0.653	0.534	0.757	0.384	0.396	0.426	0.350	0.475
`ExaRanker`	50k	0.673	0.540	0.778	0.423	0.413	0.431	0.349	0.489
`monoT5`	15k	0.656	0.523	0.746	0.392	0.382	0.409	0.344	0.466
`ExaRanker`	15k	0.683	0.531	0.752	0.403	0.408	0.415	0.352	0.477
`monoT5`	10k	0.643	0.510	0.749	0.379	0.374	0.426	0.341	0.463
`ExaRanker`	10k	0.667	0.527	0.752	0.409	0.393	0.418	0.347	0.474
`monoT5`	5k	0.625	0.488	0.693	0.364	0.337	0.417	0.328	0.438
`ExaRanker`	5k	0.665	0.505	0.750	0.389	0.380	0.414	0.345	0.464
`monoT5`	2.5k	0.611	0.486	0.666	0.334	0.328	0.370	0.325	0.418
`ExaRanker`	2.5k	0.650	0.496	0.686	0.393	0.306	0.398	0.335	0.436

The findings of our study are visually presented in Figure 19, which clearly shows that as the size of the dataset decreases, the benefits of using explanations tend to increase. The results indicate that when comparing the effectiveness of the monoT5 model finetuned on 50k positive pairs with `ExaRanker` finetuned on 10k positive pairs, the average scores are quite similar, even though `ExaRanker` has been trained with only one-fifth of the data (4x less data). This highlights the potential of data augmentation through explanations, which effectively distills knowledge from LLMs and reduces the reliance on massive datasets to achieve good effectiveness in information retrieval tasks.

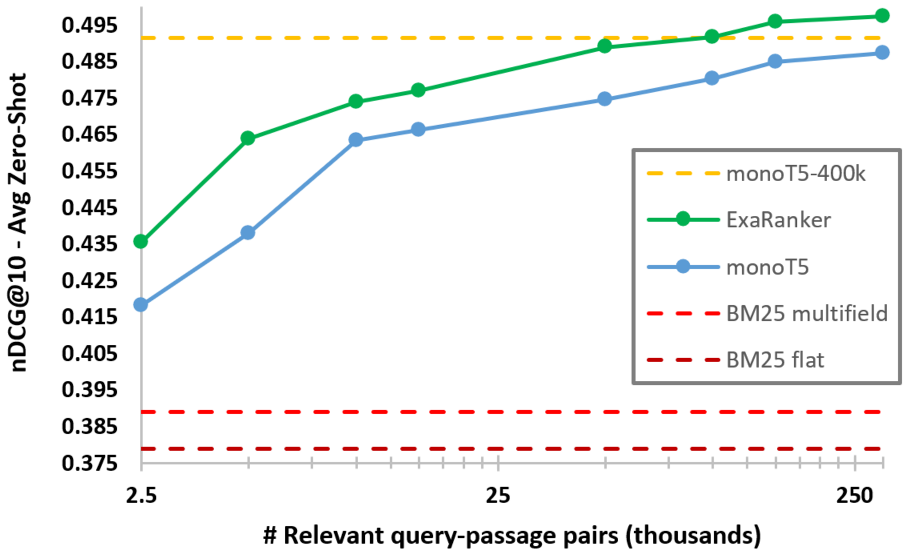

Figure 19: Average zero-shot results on 6 datasets of the BEIR benchmark. monoT5-400k is finetuned on the 400k relevant query-passage pairs from MS MARCO without explanations. Note the log scale in horizontal axis.

5.1 LLMs comparisons

The outcomes presented so far are not based on open-source LLM which leads to a cost constraint in our method. In order to address this problem, the experiment has been done using open-source LLMs to generated explanation in two different dataset sizes (as described in Table 1) which results are presented in the Table 4. This version of our model is dubbed as `ExaRanker-Open`.

Table 4: Results (nDCG@10) of open-source LLMs. Average zero-shot (all except DL 20). The column "Ft Pos." is the number of positive training examples on which the model was finetuned.

Model	**LLM**	**Ft Pos.**	**DL 20**	**Robust**	**Covid**	**Dbp**	**FiQA**	**News**	**NFC**	**Avg ZS**
monoT5	n/a	15k	0.656	0.523	0.746	0.392	0.382	0.409	0.344	0.466
`ExaRanker`	GPT-3.5	15k	0.683	0.531	0.752	0.403	0.408	0.415	0.352	0.477
`ExaRanker-Open`	Llama-2-70B	15k	0.653	0.551	0.730	0.398	0.393	0.425	0.341	0.473
`ExaRanker-Open`	Llama-2-7B	15k	0.662	0.523	0.737	0.407	0.392	0.421	0.344	0.471
monoT5	n/a	50k	0.653	0.534	0.757	0.384	0.396	0.426	0.350	0.475
`ExaRanker`	GPT-3.5	50k	0.673	0.540	0.778	0.423	0.413	0.431	0.349	0.489
`ExaRanker-Open`	Llama-2-70B	50k	0.670	0.563	0.757	0.414	0.403	0.440	0.345	0.487
`ExaRanker-Open`	Llama-2-7B	50k	0.670	0.529	0.741	0.419	0.398	0.439	0.349	0.479

For each block, the initial row represents the T5-base model finetuned without data augmentation, solely relying on categorical labels. The second row reflects the earlier results of ExaRanker using GPT-3.5. The last two rows present the outcomes of this study, employing Llama-2-70B and Llama-2-7B, respectively.

As evident, the zero-shot effectiveness is enhanced when employing a larger LLM model. Illustrated in Figure 20, the performance of Llama-2-7B surpasses that of the monoT5 without explanations in both evaluated dataset sizes, 15k and 50k relevant pairs of query-passage. However, Llama-2-70B outperforms Llama-2-7B, specially when using the larger dataset with 50k query-relevant passage pairs. Ultimately, ExaRanker, using GPT-3.5, remains the top-performing model.

These results strongly suggest the quality of data augmentation produced by each Llama model size versus GPT-3.5. As expected, larger models exhibit superior natural language processing capabilities, leading to a more substantial extraction of signals that can be effectively utilized during the finetuning phase.

Although we can question how much these LLMS are actually zero-shot due to having been trained on large scale data from different sources of information, the results are still demonstrating that the explanations leads to better results independently of the LLM size used for data augmentation. These results reinforce the fact that the text augmentation is an effective strategy to better finetune neural rerankers as the models have more signal and data to properly retrieve information and rerank the relevant passages to the queries.

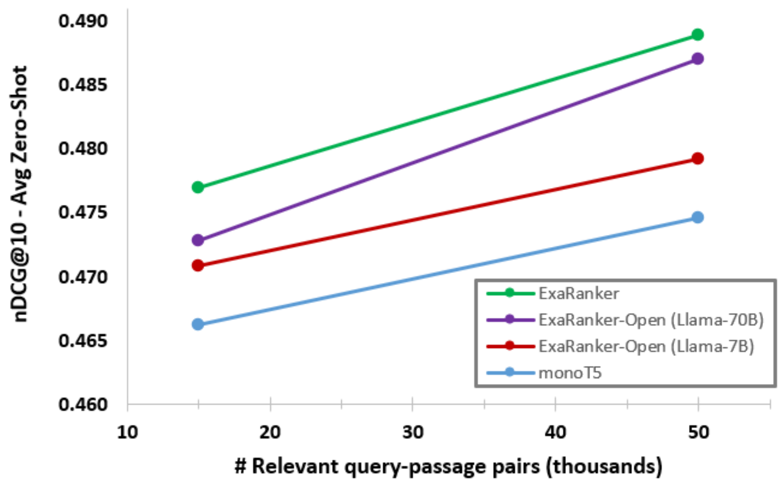

Figure 20: Average zero-shot results on 6 datasets of the BEIR benchmark with respect to training dataset size, comparing the 4 models evaluated.

5.2 Qualitative Analysis

Table 5 and Table 6 present some outputs generated by `ExaRanker` and `ExaRanker-Open` using either commercial and open-source LLMs augmented datasets. These outputs are generated from TREC-DL 2020 dataset samples, providing a qualitative comparison of the model's predictions.

Overall, the model generates reasonable explanations that help to improve understanding of the relationship between the query and the passage. The correct and incorrect explanations passages generated by the models are highlighted in red for a better assessment and visualization about the output quality got from the ExaRanker models.

For instance, in the second sample in Table 5, the models correctly predict the non-relevance of the passage even though it mentions specific terms from the query, such as “early pregnancy“. This suggests that the models were able to extract important information from the query and use it to make relevant predictions.

However, as shown in the primary example in Table 6, the models may struggle with more complex relationships, such as those between electromagnetic and radio waves which is a type of electromagnetic wave. This highlights a potential limitation of the approach, as the model's effectiveness may be limited by the complexity of the relationships it is able to identify. It also indicates the models does not carry any type of memory or reasoning in this IR application, instead, they seem to be able to use only the information available at that samples, i.e. limited to the passage and carried context.

Nevertheless, these results provide strong evidence that explanations can be used to improve the effectiveness of large language models on information retrieval tasks. By providing additional information during the training process, explanations can help the language models by reducing the amount of training data needed to achieve high levels of effectiveness. However, the models do not necessarily rely on reasoning to arrive at an answer. Instead, the explanations are used to expedite the learning process, with the most relevant score for ranking the relationship between sequences being concentrated in the label score, specifically the true or false label indicating the relevance relation, rather than in the entire explanation generated.

From a reason perspective, it is important to note that datasets generated using larger models provides more comprehensive explanations. Comparing the outputs generated by the ExaRanker trained using the GPT-3.5 in the augmented dataset versus the one using Llama, it is notably the difference in the explanation quality. While the GPT-3.5 leads to a concise and clear understand of relevance relation between query and passage, the Llama tends to hallucinate and create unreal explanations which leads to wrong relevance classification.

It can be easily noted the primary example in Table 5. Although predicting the relevance relation properly, the Llama-2-7B rewrite all the text again, including irrelevant information for the query such as “according to Bureau of Labor statistics“. In a similar problem, the Llama-2-70B is much more verbose and repeats the term “relevant to the question“ twice in the explanation. On the other hand, GPT-3.5 is direct and concise, highlighting only the key terms in the explanation.

The same situation can be observed in a wrong prediction. For instance, the second example in Table 6, we see the Llama models generating more text and repeating the term “inter-sheet“ which is not a key-word in the explanation. Also, there is a mistake in the smaller LLama model (Llama-2-7B), which has failed to predict the relation properly although providing a very similar explanation as the Llama-2-70B. In this case, the GPT-3.5 has provided a small explanation but failed in the relevance relation as well.

Relevant	Input	Is the question: **"average salary for dental hygienist in nebraska"** answered by the document: **"Dental Hygienist Salaries by Area. According to the Bureau of Labor Statistics, Dental Hygienists in the state of Nebraska can expect to earn an average of $65,236 annually or $31.36 per hour."**? Give an explanation.
	Output GPT-3.5	true. Explanation: The question is about average salary for dental hygienist in nebraska. The passage describes the average salary for dental hygienist in Nebraska.
	Output Llama-2-7B	true. Explanation: The question is about the average salary for dental hygienist in Nebraska. The passage provides information on the average salary for dental hygienists in Nebraska according to the Bureau of Labor Statistics.
	Output Llama-2-70B	true. Explanation: The question is about the average salary for a dental hygienist in Nebraska. The passage provides information on the average salary for dental hygienists in Nebraska, which is relevant to the question. Therefore, the passage is relevant to the question.
Non-relevant	Input	Is the question: **"can fever cause miscarriage early pregnancy¿'** answered by the document: **"Pregnancy Early symptoms of pregnancy include a missed period, morning sickness, breast swelling, fatigue, and more. Chickenpox (varicella) Chickenpox is an illness that causes a red, itchy rash, fever, headache, sore throat, and body aches"**? Give an explanation.
	Output GPT-3.5	false. Explanation: The question is about fever causing miscarriage early pregnancy. The passage is about chickenpox causing fever.
	Output Llama-2-7B	false. Explanation: The question is about whether fever can cause miscarriage early in pregnancy. The passage is about early symptoms of pregnancy, including chickenpox, which is a different illness.
	Output Llama-2-70B	false. Explanation: The question is about whether fever can cause miscarriage early in pregnancy. The passage mentions chickenpox, which is an illness that causes fever, but it does not address the question of whether fever can cause miscarriage early in pregnancy. Therefore, the passage is not relevant to the question.

Table 5: Outputs generated by the `ExaRanker` model - mostly correct prediction.

Relevant	Input	Is the question: **“what medium do radio waves travel through“** answered by the document: **“An electromagnetic wave (this includes light) does not require a physical medium - it can travel through empty space.“**? Give an explanation.
	Output GPT-3.5	false. Explanation: The question is about the medium of radio waves. The passage is about electromagnetic waves.
	Output Llama-2-7B	false. Explanation: The question is about the medium through which radio waves travel. The passage is about electromagnetic waves and their ability to travel through empty space.
	Output Llama-2-70B	false. Explanation: The question is about the medium through which radio waves travel. The passage does not mention radio waves or their medium, but rather discusses electromagnetic waves and their ability to travel through empty space. Therefore, the passage is not relevant to the question.
Non-relevant	Input	Is the question: **“do google docs auto save“** answered by the document: **“Inter-sheet formulas in Google Docs are similar to those used in Microsoft Excel. Using this ability, you can post a value automatically in a cell on one tab that is calculate or posted in a cell on another tab.“**? Give an explanation.
	Output GPT-3.5	true. Explanation: The question is about google docs auto save. The passage describes how google docs auto save.
	Output Llama-2-7B	true. Explanation: The question is about Google Docs auto save. The passage explains how Google Docs uses inter-sheet formulas to post values automatically in cells on one tab, which is relevant to the question.
	Output Llama-2-70B	false. Explanation: The question is about auto-save in Google Docs, but the passage only discusses inter-sheet formulas in Google Docs and their similarity to Microsoft Excel. It does not provide any information about auto-save or the ability to save data. Therefore, the passage is not relevant to the question.

Table 6: Outputs generated by the `ExaRanker` model - mostly incorrect prediction (except by Llama-2-70b at non-relevant query-passage case).

5.3 Ablation Experiments

We conducted two supplementary experiments to gain a better understanding of the impact of incorporating explanations as training objectives. In the first experiment, we finetuned another `ExaRanker` model for 30 epochs, using the same hyperparameters, but with the label and explanation generation order inverted. We expected this approach to lead to better results, as the model would first elaborate on the relationship between the query and the passage before predicting a label.

However, as shown in Table 7, the nDCG@10 score decreased by 9.9 points compared to the original method of generating a relevance label followed by an explanation. This outcome may be counterintuitive and contradicts previous findings in chain-of-thought researches [**?**].

This result is likely due to the difficulty of consolidating the probabilities of multiple generated tokens into a single ranking score. We explored several approaches, such as using the average token probabilities of the entire sequence or only the probability of the final token (`true` or `false`), but in all cases, the nDCG@10 score was lower than the approach described in Section 3. Also, it would impact in the processing time of the reranker as the entire explanation would need to be generated to rank the documents which is time consuming and would make it impossible to use this type of solution in the IR field.

Output template	**nDCG@10**
`{label}. Explanation: {explanation}`	0.683
`Explanation: {explanation}. {label}.`	0.584

Table 7: Ablation of the output template. Results on TREC-DL 2020 with models finetuned on 15k pos. + 15k neg. samples.

The second ablation study conducted was done to investigate if explanations can improve a model that has already been finetuned on a large ranking dataset. For this purpose, we further finetuned a monoT5-base model, which had already been trained on 400k positive pairs from MS MARCO, on 15k positive and 15k negative examples with explanations. The results in Table 8 indicate a minimal difference of 0.2 nDCG@10 points on average over the 7 datasets. As previously discussed, the benefits of explanations appear to diminish when a large training dataset is available. This experiment also demonstrates that finetuning with explanations does not undermine the effectiveness of a ranker while providing it with the ability to generate explanations.

Model	**DL 20**	**Robust**	**Covid**	**Dbp**	**FiQA**	**News**	**NFC**	**Avg**
monoT5 (ft on 400k pos)	0.652	0.536	0.777	0.419	0.413	0.447	0.357	0.514
`ExaRanker` (from monoT5)	0.701	0.528	0.756	0.398	0.406	0.442	0.352	0.512

Table 8: Results on finetuning `ExaRanker` from a monoT5-base model finetuned on 400k positive examples from MS MARCO.

6 Conclusion

This master's thesis has explored the application of large language models in information retrieval. This master thesis has explored the concept of integrating natural language explanations as supplemental labels within neural rankers to improve their performance across different datasets.

Through experimentation involving various language models, dataset sizes, and training methods, the study introduced ExaRanker, a novel approach to IR that utilizes explanations generated by LLMs to enhance ranking models. ExaRanker demonstrated superiority over traditional methods in multiple experiments, showcasing its potential to enhance search engine performance.

The study represents a shift in traditional methodologies by seamlessly combining the capabilities of LLMs with the interpretability of explanations. By enriching datasets with contextual explanations, ExaRanker improves the training process, addressing challenges such as data scarcity and linguistic nuances.

The proposed data augmentation technique automates explanation generation, reducing the need for manual annotation while enhancing the richness of training data. ExaRanker consistently outperformed baseline models across diverse datasets and evaluation metrics without sacrificing speed, making it suitable for real-world applications.

Notably, experiments with smaller datasets underscored ExaRanker's efficiency in utilizing limited training data, highlighting the effectiveness of explanations as a form of augmentation during the finetuning process of neural rankers. Additionally, comparisons of different LLMs emphasized the importance of model size in capturing language patterns, while confirming the consistent efficacy of explanations across different LLM sizes.

The thesis contributes to IR by demonstrating the integration of natural language explanations into neural rankers and paving the way for its application in diverse domains. By enhancing retrieval models through automated explanation augmentation, the study improves effectiveness while reducing data requirements, ultimately leading to more efficient search engines. It advances our understanding of utilizing language processing techniques for IR challenges while emphasizes the significance of incorporating natural language explanations in the training process to improve model generalization and effectiveness across different datasets.

Beyond the direct contributions to the IR field, the proposed methodology holds promise for extension and application in diverse areas and objectives beyond data retrieval. Further research could explore additional strategies for optimizing the use of explanations in IR tasks, as well as investigate the scalability and applicability of ExaRanker in real-world search scenarios.

6.1 Future Work

Some paths emerge from of our study with ExaRanker and LLMs application in the IR field. These avenues offer a continued exploration of text generation and its benefits to expand the applications of ExaRanker as briefly discussed in this section.

One avenue for future exploration involves extending the application of ExaRanker to different domains and applications. By adapting and finetuning the methodology to suit diverse contexts such as document classification, sentiment analysis, or recommendation systems, we can unlock new possibilities for leveraging explanations in various real-world scenarios. For example, the method could find application in decision-making processes across industries like retail, healthcare, or finance, offering valuable insights and augmenting human judgment in critical areas such as product recommendation, medical diagnosis, or financial risk assessment.

Another direction for future research lies in exploring the ethical implications and potential applications of ExaRanker in sensitive areas. Given that the method generates explanations that elucidate the underlying rationale behind decision-making processes, there arises an opportunity to investigate its use in contexts such as social decisions, legal judgments, or policy formulation. By addressing ethical considerations and ensuring transparency and accountability in decision-making, ExaRanker could contribute to enhancing fairness, equity, and accountability in high-stakes decision-making contexts. Although a final human decision would be done, the method proposed by our study could explain and rationalize any recommend decision made by an AI system or integrated application.

Furthermore, exploring the listwise application of ExaRanker presents an interesting avenue for future investigation. In this approach, a top-N list of ranked documents serves as input, and the explanations generated by the model are utilized to inform decision-making and refine the ranking order. By leveraging the rich contextual information encoded in the explanations, ExaRanker could facilitate more nuanced and informed decision-making processes, particularly in scenarios where document ranking is crucial, such as content recommendation, search engine result ranking, or information triage in large-scale document repositories. It also could improve the performance when the list of documents is limited as the ExaRanker method showcase good results even with data scarcity scenario.

References

[1] *SIGIR '23: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2023. Association for Computing Machinery.

[2] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268v3*, 2018.

[3] H. Bast and M. Celikik. Efficient index-based snippet generation. *ACM Transactions on Information Systems (TOIS)*, 32(2):1–24, 2014.

[4] L. Bonifacio, H. Abonizio, M. Fadaee, and R. Nogueira. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392, 2022.

[5] V. Boteva, D. Gholipour, A. Sokolov, and S. Riezler. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*, pages 716–722. Springer, 2016.

[6] W.-F. Chen, S. Syed, B. Stein, M. Hagen, and M. Potthast. Abstractive snippet generation. In *Proceedings of The Web Conference 2020*, pages 1309–1319, 2020.

[7] N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the TREC 2020 deep learning track. *CoRR*, abs/2102.07662, 2021.

[8] Z. Dai and J. Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '19. ACM, July 2019.

[9] Z. Dai, C. Xiong, J. Callan, and Z. Liu. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 126–134, New York, NY, USA, 2018. Association for Computing Machinery.

[10] Z. Dai, V. Y. Zhao, J. Ma, Y. Luan, J. Ni, J. Lu, A. Bakalov, K. Guu, K. B. Hall, and M.-W. Chang. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*, 2022.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, June 2019. Association for Computational Linguistics.

[12] P. Fernandes, M. Treviso, D. Pruthi, A. F. Martins, and G. Neubig. Learning to scaffold: Optimizing model explanations for teaching. *arXiv preprint arXiv:2204.10810*, 2022.

[13] Z. T. Fernando, J. Singh, and A. Anand. A study on the interpretability of neural retrieval models using deepshap. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1005–1008, 2019.

[14] F. Ferraretto, T. Laitz, R. Lotufo, and R. Nogueira. Exaranker: Explanation-augmented neural ranker, 2023.

[15] F. Ferraretto, T. Laitz, R. Lotufo, and R. Nogueira. Exaranker: Synthetic explanations improve neural rankers. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2409–2414, New York, NY, USA, 2023. Association for Computing Machinery.

[16] F. Ferraretto, T. Laitz, R. Lotufo, and R. Nogueira. Exaranker-open: Synthetic explanation for ir using open-source llms, 2024.

[17] T. Formal, B. Piwowarski, and S. Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292, 2021.

[18] L. Gao and J. Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[19] L. Gao, Z. Dai, and J. Callan. Rethink training of bert rerankers in multi-stage retrieval pipeline. *arXiv preprint arXiv:2101.08751*, 2021.

[20] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM'16. ACM, Oct. 2016.

[21] F. Hasibi, F. Nikolaev, C. Xiong, K. Balog, S. E. Bratsberg, A. Kotov, and J. Callan. Dbpedia-entity v2: a test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1265–1268, 2017.

[22] S. Hofstätter, O. Khattab, S. Althammer, M. Sertkan, and A. Hanbury. Introducing neural bag of whole-words with colberter: Contextualized late interactions using enhanced reduction. *arXiv preprint arXiv:2203.13088*, 2022.

[23] S. Hofstätter, S.-C. Lin, J.-H. Yang, J. Lin, and A. Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2021.

[24] J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

[25] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning, 2021.

[26] V. Jeronymo, L. Bonifacio, H. Abonizio, M. Fadaee, R. Lotufo, J. Zavrel, and R. Nogueira. Inpars-v2: Large language models as efficient dataset generators for information retrieval, 2023.

[27] U. Katz, M. Geva, and J. Berant. Inferring implicit relations with language models. *arXiv preprint arXiv:2204.13778*, 2022.

[28] C. Li, A. Yates, S. MacAvaney, B. He, and Y. Sun. Parade: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093*, 2020.

[29] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira. Pyserini: An easy-to-use Python toolkit to support replicable ir research with sparse and dense representations. *ArXiv*, abs/2102.10073, 2021.

[30] J. Lin, R. Nogueira, and A. Yates. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325, 2021.

[31] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[32] S. Lu, D. He, C. Xiong, G. Ke, W. Malik, Z. Dou, P. Bennett, T.-Y. Liu, and A. Overwijk. Less is more: Pretrain a strong Siamese encoder for dense text retrieval using a weak decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2791, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[33] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104, 2019.

[34] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur. Www'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1941–1942, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[35] N. Moniz, L. Torgo, and J. Vinagre. Data-driven relevance judgments for ranking evaluation, 2016.

[36] S. Mysore, A. Cohan, and T. Hope. Multi-vector models with textual guidance for fine-grained scientific document similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4453–4470, Seattle, United States, July 2022. Association for Computational Linguistics.

[37] R. Nogueira and K. Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[38] R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, 2020.

[39] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[40] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[41] R. Rahimi, Y. Kim, H. Zamani, and J. Allan. Explaining documents' relevance to search queries. *arXiv preprint arXiv:2111.01314*, 2021.

[42] G. Recchia. Teaching autoregressive language models complex tasks by demonstration. *arXiv preprint arXiv:2109.02102*, 2021.

[43] D. Reinsel, J. Gantz, and J. Rydning. The digitization of the world from edge to core. an idc white paper-us44413318, sponsored by seagate., Nov 2018.

[44] K. Roberts, T. Alam, S. Bedrick, D. Demner-Fushman, K. Lo, I. Soboroff, E. Voorhees, L. L. Wang, and W. R. Hersh. TREC-COVID: rationale and structure of an information re-

trieval shared task for COVID-19. *Journal of the American Medical Informatics Association*, 27(9):1431–1436, 07 2020.

[45] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[46] J. Rorseth, P. Godfrey, L. Golab, M. Kargar, D. Srivastava, and J. Szlichta. Credence: Counterfactual explanations for document ranking, 2023.

[47] G. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira. In defense of cross-encoders for zero-shot retrieval. *arXiv preprint arXiv:2212.06121*, 2022.

[48] G. M. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira. No parameter left behind: How distillation and model size affect zero-shot retrieval. *arXiv preprint arXiv:2206.02873*, 2022.

[49] D. Roy, S. Saha, M. Mitra, B. Sen, and D. Ganguly. I-rex: a lucene plugin for explainable ir. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2949–2952, 2019.

[50] D. S. Sachan, M. Lewis, M. Joshi, A. Aghajanyan, W.-t. Yih, J. Pineau, and L. Zettlemoyer. Improving passage retrieval with zero-shot question generation. 2022.

[51] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics.

[52] P. Sen, D. Ganguly, M. Verma, and G. J. Jones. The curious case of ir explainability: Explaining document scores within and across ranking models. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2069–2072, 2020.

[53] J. Singh and A. Anand. Interpreting search result rankings through intent modeling. *arXiv preprint arXiv:1809.05190*, 2018.

[54] J. Singh and A. Anand. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 770–773, 2019.

[55] I. Soboroff, S. Huang, and D. Harman. Trec 2018 news track overview.

[56] N. Thakur, N. Reimers, and J. Lin. Domain adaptation for memory-efficient dense retrieval. *arXiv preprint arXiv:2205.11498*, 2022.

[57] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. 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)*, 2021.

[58] P. Thomas, B. Billerbeck, N. Craswell, and R. W. White. Investigating searchers' mental models to inform search explanations. *ACM Transactions on Information Systems (TOIS)*, 38(1):1–25, 2019.

[59] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10, 1998.

[60] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[61] A. Turpin, Y. Tsegay, D. Hawking, and H. E. Williams. Fast generation of result snippets in web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 127–134, 2007.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[63] M. Verma and D. Ganguly. Lirme: locally interpretable ranking model explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1281–1284, 2019.

[64] M. Völske, A. Bondarenko, M. Fröbe, B. Stein, J. Singh, M. Hagen, and A. Anand. Towards axiomatic explanations for neural ranking models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 13–22, 2021.

[65] E. Voorhees. Overview of the trec 2004 robust retrieval track, 2005-08-01 2005.

[66] K. Wang, N. Thakur, N. Reimers, and I. Gurevych. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *arXiv preprint arXiv:2112.07577*, 2021.

[67] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[68] X. Xie, Q. Dong, B. Wang, F. Lv, T. Yao, W. Gan, Z. Wu, X. Li, H. Li, Y. Liu, and J. Ma. T2ranking: A large-scale chinese benchmark for passage ranking, 2023.

[69] J. Xin, C. Xiong, A. Srinivasan, A. Sharma, D. Jose, and P. N. Bennett. Zero-shot dense retrieval with momentum adversarial domain invariant representations. *arXiv preprint arXiv:2110.07581*, 2021.

[70] P. Yu, R. Rahimi, and J. Allan. Towards explainable search results: A listwise explanation generator. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 669–680, 2022.

[71] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. Chi. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

[72] H. Zhuang, Z. Qin, R. Jagerman, K. Hui, J. Ma, J. Lu, J. Ni, X. Wang, and M. Bendersky. Rankt5: Fine-tuning t5 for text ranking with ranking losses. *arXiv preprint arXiv:2210.10634*, 2022.

[73] H. Zhuang, X. Wang, M. Bendersky, A. Grushetsky, Y. Wu, P. Mitrichev, E. Sterling, N. Bell, W. Ravina, and H. Qian. Interpretable ranking with generalized additive models. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 499–507, 2021.