Explanation and Judgement of IR Ranking using LLM

Dissertation submitted in partial fulfilment of the requirements
for the award of the degree of

Master of Technology

by

Santanu Mondal

(Roll No. CS2319)

Under the Supervision of

Dr. Debapriyo Majumdar

INDIAN STATISTICAL INSTITUTE KOLKATA

Kolkata, India

June, 2025

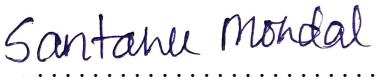
CERTIFICATE

This is to certify that the dissertation titled **"Explanation and Judgement of IR Ranking using LLM"**, submitted by **Santanu Mondal** to the Indian Statistical Institute, Kolkata, as part of the requirements for the Master of Technology in Computer Science, is a true record of the work done by him under my supervision. The dissertation meets the necessary requirements set by the institute.

Dr. Debapriyo Majumdar
CVPR Unit
Indian Statistical Institute
Kolkata - 700108
India

Declaration

I confirm that this dissertation is based on my own work and ideas. Wherever I have used the ideas or words of others, I have given proper credit through citations and references. I have followed the principles of academic honesty and integrity, and I have not misrepresented, falsified, or fabricated any data, facts, or sources. I understand that breaking these rules may lead to disciplinary action by the Institute or legal action from the original source if proper credit or permission was not given.



Santanu Mondal
Roll No.: CS2319
Place: ISI Kolkata

Acknowledgements

I would like to sincerely thank Dr. Debapriyo Majumdar, my advisor at the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, for his valuable guidance, constant support, and encouragement throughout this work. His deep knowledge and insightful feedback have greatly helped me learn and grow as a researcher.

I am also thankful to all the faculty members at the Indian Statistical Institute for their helpful suggestions and teachings, which added important perspective to my research. I especially thank Mr. Sourav Saha for his help in preparing this thesis.

I am truly grateful to my parents and family for their constant support, and to my friends for their encouragement and help along the way. Finally, I thank everyone who contributed to this journey, even if not mentioned by name here.

Abstract

Pretrained transformer models such as BERT and T5 have significantly advanced the performance of information retrieval (IR) systems when fine-tuned with large-scale labeled datasets. However, their effectiveness diminishes notably in low-resource scenarios where annotated query-passage pairs are limited. This thesis explores an alternative supervision strategy by leveraging natural language explanations to enhance training signals during fine-tuning. We propose a novel methodology that augments traditional relevance labels with textual explanations generated by a large language model (LLM) using few-shot prompting.

To achieve this, we generate explanations for 30,000 query-passage-label triples from the MS MARCO dataset using the open-source model `google/gemma-2b`, allowing for cost-free and scalable inference. These augmented samples are then used to fine-tune a T5-base sequence-to-sequence model, with the objective of producing both the relevance label and an accompanying explanation. During inference, the model predicts the label token, and the probability of that token is used as a soft relevance score, enabling efficient ranking.

Empirical results demonstrate that our explanation-augmented retriever outperforms strong baselines, including BM25, a BERT reranker, and a T5 model trained with labels only. We further analyze the effectiveness of explanation order, training data size, and the quality of generated rationales. Our findings suggest that natural language explanations offer a powerful form of supervision, particularly valuable in data-scarce IR settings, and present a compelling direction for improving neural retrievers with minimal annotation overhead.

Keywords: Information Retrieval, Natural Language Explanations, Large Language Models, T5, MS MARCO, Sequence-to-Sequence Learning, Fine-Tuning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Recent advancements in pre-trained transformer models such as BERT (Devlin et al., 2019) and T5 (Nogueira et al., 2020) have led to substantial improvements in various information retrieval (IR) tasks when these models are fine-tuned on large-scale labeled datasets. When fine-tuned on hundreds of thousands of query-passage examples, these models often outperform traditional retrieval methods, particularly in settings where the test data closely resembles the training distribution. For instance, monoT5, a reranking model trained on 400,000 positive query-passage pairs from the MS MARCO dataset, significantly surpasses BM25 on 15 out of the 18 datasets included in the BEIR benchmark, showcasing the strength of supervised transformer-based retrieval systems.

However, this impressive performance comes with a caveat: these models rely heavily on the availability of large-scale annotated data. In scenarios where only limited supervision is available, their effectiveness drops notably. As an example, a BERT-based reranker trained on only 10,000 labeled query-passage pairs shows marginal

improvements over BM25 on the MS MARCO passage ranking task. While researchers have attempted to mitigate the dependency on extensive labeled data by increasing model capacity or incorporating pretraining objectives specifically designed for IR, these solutions often demand significant computational resources, which may not be feasible for all practitioners.

One of the core challenges in training neural retrieval models with limited data stems from the nature of the supervision they receive. Typically, models are fine-tuned using categorical labels such as "relevant" or "non-relevant." These binary labels, while straightforward, provide minimal insight into the rationale behind a passage's relevance, making it harder for models to learn deeper semantic relationships. This scenario is akin to training a human evaluator to judge relevance using only yes/no answers without providing any reasoning. Naturally, the learning process would be more effective and intuitive if detailed natural language explanations accompanied each example, offering richer supervision.

To address this issue, we propose an approach that leverages natural language explanations as supplementary training signals for retrieval models. By incorporating explanations that justify the relevance or irrelevance of a passage to a query, our method enhances model training and reduces the reliance on large quantities of labeled data. Moreover, we demonstrate that few-shot large language models (LLMs) like GPT-3.5 can be utilized to automatically generate such explanations, enabling the augmentation of existing datasets without requiring labor-intensive manual annotations. This makes our approach more accessible and scalable across different IR tasks and domains.

Our experiments reveal two key findings. First, the benefit of adding natural language explanations is most pronounced in low-data regimes, and diminishes as the

volume of supervised data increases. Second, the ordering of explanation and label generation plays a critical role: training the model to first predict the label and then generate the corresponding explanation yields better performance than doing the reverse. Interestingly, this outcome challenges prior assumptions observed in chain-of-thought prompting studies, where explanations often precede answers. These results suggest new directions for designing more effective training strategies in neural information retrieval.

1.2 Problem Statement

Despite the remarkable success of pretrained transformer models such as BERT and T5 in information retrieval (IR) tasks, their performance is heavily reliant on the availability of large-scale labeled datasets. These models exhibit high effectiveness when fine-tuned on datasets containing hundreds of thousands of annotated query-passage pairs. However, in real-world applications, such extensive labeled data is rarely available due to the high cost and time required for manual annotation. As a result, the deployment of powerful neural retrieval models is often restricted to well-resourced domains, limiting their broader applicability.

The core issue lies in the nature of the supervision used during fine-tuning. Most IR models are trained using categorical labels—typically binary indicators of relevance (e.g., relevant or non-relevant). While these labels are easy to obtain in principle, they convey very limited information. They do not capture the underlying semantic reasoning or contextual alignment between the query and the passage. This form of weak supervision makes it difficult for the model to understand why a document is relevant to a query, thereby requiring an abundance of examples for effective learning.

This challenge is exacerbated in out-of-domain or low-resource settings where the labeled data available for fine-tuning is scarce. In such scenarios, models trained with only categorical feedback perform only marginally better than traditional unsupervised retrieval techniques like BM25. This suggests that the current supervision paradigm is not only data-hungry but also inefficient in terms of the quality of supervision it provides.

Furthermore, increasing model size or using IR-specific pretraining objectives has been proposed as a workaround for limited data. However, these strategies come with trade-offs—namely, increased computational complexity, higher energy consumption, and a steeper barrier to entry for researchers and practitioners with constrained resources. Thus, there is a need for alternative methods that enhance the data efficiency of training neural retrieval models without significantly increasing their computational demands.

We hypothesize that one of the reasons for the inefficiency of current fine-tuning approaches is the lack of rich, informative supervision. Learning from binary labels alone is akin to training a human with yes/no feedback, devoid of any explanatory context. A more informative learning process would involve access to explanations—descriptive annotations that clarify the reasoning behind a passage's relevance to a given query. Such explanations can serve as a form of rational supervision, helping the model to internalize the semantic and logical relationships between queries and documents.

However, generating natural language explanations for IR tasks poses its own challenges. Manual explanation creation is labor-intensive and not scalable. Recent advances in large language models (LLMs) such as GPT-3.5 offer a promising alternative. These models can generate high-quality explanations in a few-shot setting,

providing a feasible pathway for augmenting training data with natural language rationales at scale.

Given this context, the problem we aim to address in this work is threefold:

1. How can we enhance the training of neural IR models in low-data regimes by supplementing weak categorical labels with natural language explanations?

2. Can few-shot prompting of LLMs be effectively used to generate relevant and coherent explanations for query-passage pairs without manual effort?

3. What is the optimal training strategy for incorporating these explanations to maximize retrieval performance, and how does it compare with standard fine-tuning methods?

By addressing these questions, our goal is to develop a training framework that enables more data-efficient learning for neural retrieval models. This involves designing methods that not only perform well with limited supervision but also generalize across domains with minimal additional annotation cost. Ultimately, we aim to democratize access to high-performing IR systems by reducing their dependency on large-scale manually labeled datasets and making them more practical for diverse real-world applications.

1.3 Our Contribution

In this thesis, we aim to reproduce and extend the work proposed by Ferraretto et al. (2023), where natural language explanations are used to enhance the performance of information retrieval models. The original study employed GPT-3.5, a powerful proprietary large language model, to generate relevance explanations for

query-passage pairs. While effective, the use of GPT-3.5 led to substantial inference costs—amounting to approximately 840 USD—making the approach less accessible and less scalable for widespread academic or low-resource adoption.

Our primary contribution lies in implementing the same methodology using only open-source tools. Specifically, we substitute GPT-3.5 with the `google/gemma-2b` model available through the Hugging Face Transformers library. This decision significantly reduces the computational expense, as `gemma-2b` can be run on local hardware or cloud platforms with minimal cost, enabling cost-free inference for generating explanations. By doing so, we democratize access to this line of research and make it more reproducible and practical for the broader IR and NLP research community.

In addition, we slightly refined the prompt used for explanation generation. Our revised prompt structure improves clarity and relevance alignment between the query and passage. The modified prompt emphasizes concise reasoning grounded in the content of the passage and helps elicit more focused and interpretable explanations from the model.

Through these contributions, we demonstrate that high-quality explanation-based retrieval systems can be built without relying on commercial LLMs, thereby offering a more sustainable and accessible alternative for research and deployment.

1.4 Structure of the Thesis

This thesis is organized into six chapters, each of which builds upon the previous to provide a comprehensive overview of the work undertaken:

- **Chapter 1: Introduction and Problem Statement**

  This chapter introduce the background and motivation for the work, highlighting the limitations of current information retrieval models and defining the specific problem addressed in this thesis.

- **Chapter 2: Related Work**

  This chapter contains a review of related work in the field of explanation-based retrieval, large language models, and methods that use natural language rationales to improve model performance.

- **Chapter 3: Our Approach**

  This chapter outlines the overall approach taken in this thesis, including explanation generation using open-source LLMs, prompt design, and model finetuning for retrieval.

- **Chapter 4: Dataset Description**

  This chapter gives the description of the datasets used for explanation generation and model training, particularly the MS MARCO passage ranking dataset.

- **Chapter 5: Analysis and Results**

  This chapter presents the experimental setup, performance evaluation, and analysis of the results obtained from the finetuned retrieval models.

- **Chapter 6: Conclusion and Future Work**

  The final chapter first summarizes the main contribution of this work and then discusses possible directions for future research in the area of explanation-augmented retrieval.

Chapter 2

Related Work

A growing body of research has demonstrated that prompting large language models (LLMs) to generate step-by-step rationales in natural language—often referred to as chain-of-thought reasoning—can significantly enhance performance across a variety of complex tasks that require reasoning (Recchia, Fernandes et al., Wang et al., Zelikman et al.). These rationales act as intermediate explanations that guide the model toward better decision-making. However, most of these studies have relied on extremely large models with billions of parameters, such as GPT-3 with 175B parameters. While effective, such models are computationally intensive and therefore not practical for many information retrieval (IR) tasks.

For instance, reranking just 100 candidate passages for a single query using a 175B-parameter model may take over a minute, even when run on multiple high-end GPUs (e.g., four A100s). This makes the direct use of such models in real-time IR systems infeasible. Motivated by this limitation, our work proposes a method to distill the reasoning capabilities of these large models into smaller, more efficient retrieval models. Instead of using large LLMs during inference, we leverage them during training to enhance supervision signals—specifically by generating natural

language explanations—and transfer their knowledge to lightweight models suitable for practical deployment.

The approach discussed in this thesis is closely related to prior works such as InPars (Bonifacio et al., Jeronymo et al.), Promptagator (Dai et al., 2022), and UPR (Sachan et al., 2023). These methods also utilize large language models to improve IR systems. However, unlike InPars and Promptagator—which primarily focus on query generation from documents—we take a different route by enriching existing training labels with additional explanatory information that better captures query-document relevance. While our method differs in focus, we view these techniques as complementary and believe future work could explore integrating query generation with explanation-based supervision for even greater performance gains.

In parallel, numerous studies have explored generating explanations for search results. For example, GenEx by Rahimi et al. (2021) produces short, meaningful noun phrases (e.g., "impacts on Medicare tax") to highlight key aspects of a query-document pair. Similarly, snippet generation approaches aim to present brief excerpts that help users understand why a particular result is retrieved. These explanations are generally intended for user-facing applications—designed to enhance interpretability, transparency, or trust in the IR system.

In contrast, our work takes a different perspective: we are not focused on explaining the results to users, but rather on using explanations to train better models. We use natural language rationales during training to provide richer supervision, which in turn improves the model's retrieval effectiveness. Although our method may incidentally produce outputs that are interpretable, we do not claim interpretability as an explicit goal. We have not evaluated the factual accuracy or correctness of the generated explanations in that context. Instead, our objective is to demonstrate that

incorporating explanations during training—regardless of their interpretability—can lead to more effective retrievers.

Chapter 3

Our Approach

Our proposed approach aims to enhance information retrieval effectiveness by training a model not only on query-passage relevance labels but also on natural language explanations that justify those labels. This section outlines the entire pipeline, from explanation generation to model training and inference.

3.1 Overview

The overall framework of our method is illustrated in Figure 1, which is taken from the original paper Ferraretto et al. (2023). At a high level, the process involves three stages:

Explanation Generation: Using a large language model (LLM) to generate natural language justifications for the relevance or non-relevance of query-passage pairs.

Data Augmentation and Model Fine-tuning: Augmenting the training data with these explanations and fine-tuning a sequence-to-sequence model on the enriched dataset. Inference: Estimating relevance at test time using the fine-tuned model,

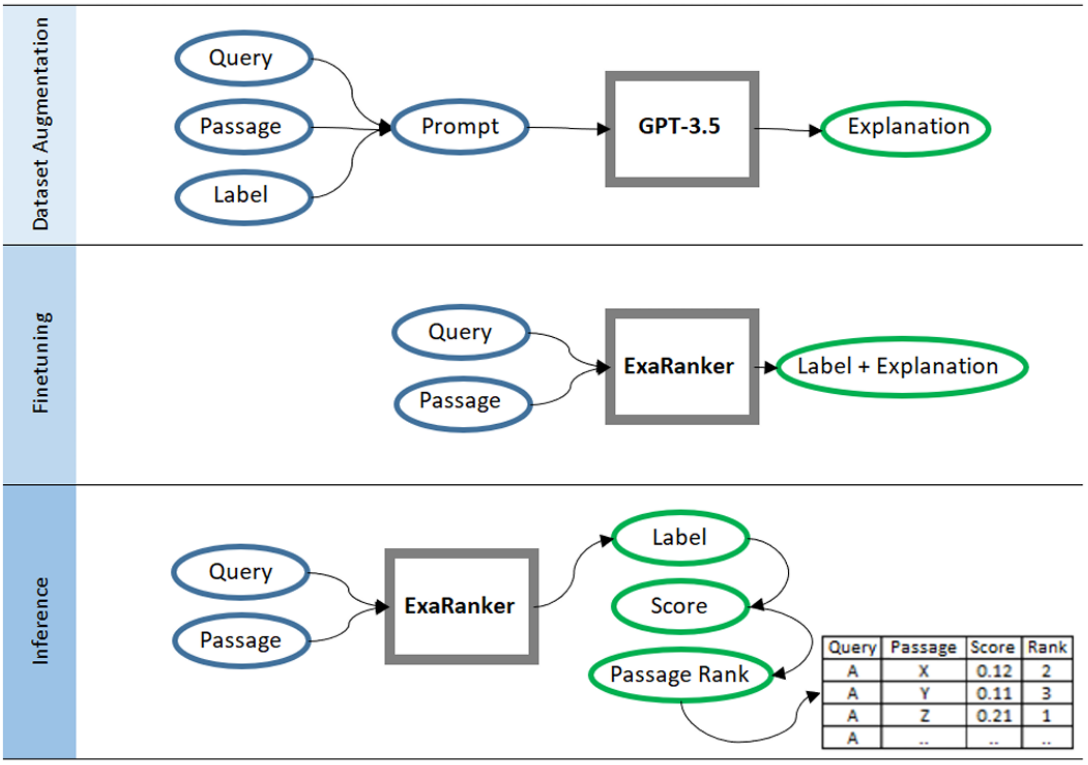


FIGURE 3.1: Overall framework of the method

relying on the likelihood of predicted labels.

3.2 Data Preparation

We begin with the MS MARCO passage ranking dataset, a widely used benchmark in IR research. From its training split, we randomly sample:

15,000 relevant query-passage pairs, and

15,000 non-relevant query-passage pairs, resulting in a balanced training set of 30,000 labeled examples.

Each sample in this dataset includes a query, a passage, and a binary relevance label (e.g., “relevant” or “not relevant”).

3.3 Generating Explanations Using an LLM

Creating human-written explanations for all 30,000 examples would be prohibitively expensive and time-consuming. To address this, we use the open-source large language model google/gemma-2b to automatically generate explanations.

Prompt Format: We craft a few-shot prompt consisting of seven illustrative examples taken from the MS MARCO dataset. Each example includes a query, a passage, a relevance label, and a natural language explanation which justifying the label.

Prompt Length: The total prompt, including instructions and examples, averages 1,400 tokens.

Inference Setup: We use greedy decoding (i.e., selecting the most probable next token at each step) and limit the generated output to 256 tokens per explanation. This keeps the explanations concise and efficient for model training.

By using google/gemma-2b, which is freely available and open source, we generate high-quality explanations without incurring additional computational or API-related costs.

3.4 Augmenting the Training Data

Each of the 30,000 training instances is then augmented to form a structured input-output sequence. Specifically:

The input consists of the query and passage.

The target output is a sequence starting with the binary label, immediately followed by the generated explanation.

This format ensures that the model learns not only to predict relevance but also to justify its prediction.

3.5 Model Fine-tuning

To ensure that the model generates coherent and contextually appropriate explanations, we include the relevance label (either true or false) in the input prompt during the explanation generation stage. This design choice is based on preliminary experiments, which revealed that explanations produced without the label were less consistent and occasionally ambiguous. By providing the label upfront, we guide the language model to focus solely on why the passage is or is not relevant—without being distracted by the task of inferring the label itself.

3.5.1 Input-Output Format for Fine-tuning

Once we generate the 30,000 explanations using the LLM, we construct training samples for fine-tuning a sequence-to-sequence model. Each training instance is formatted as a natural language prompt that asks the model to explain the relevance of a passage to a query. The specific templates used for model input and output are as follows:

Input Prompt: "Is the question query answered by the passage? Give an explanation."

Target Output: "label. Explanation: explanation"

Here, query and passage are taken directly from the MS MARCO dataset, label is true or false depending on the relevance annotation, and explanation is the corresponding natural language justification generated by the LLM in the earlier stage.

This input-output structure trains the model to first emit a binary relevance label, followed immediately by an explanation supporting that label. This format promotes learning both the classification decision and the rationale behind it.

3.5.2 Model Architecture and Training Setup

We use the T5-base model as the backbone for our fine-tuning process. Training is carried out for 15 epochs and using the AdamW optimizer, with the learning rate set to 3e-5, a weight decay of 0.01, and a batch size of 8 samples.

To handle variable-length input and output sequences, we restrict both input and output lengths to a maximum of 512 tokens. Any sequences exceeding this limit are truncated at the time of training and inference, ensuring consistent memory usage and efficiency.

For comparison, we also fine-tune a separate T5-base model on the same data and hyperparameters, but without including the explanations in the target outputs. This baseline allows us to isolate the effect of explanation-augmented training on retrieval performance.

3.6 Inference Strategy

During evaluation, we use the fine-tuned model to perform passage ranking. Given a query-passage pair, the model receives the same input format as during training,

but we limit the output to only the first token, which corresponds to the predicted label (true or false). This allows us to extract a relevance score without requiring the generation of the full explanation, significantly reducing inference time.

The relevance score s for a query-passage pair is computed based on the model's softmax probability p_0 of generating the first output token t_0:

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

This scoring mechanism assigns higher values to confident "true" predictions and penalizes confident "false" predictions, effectively ranking passages by their estimated relevance.

Since the model is trained with a causal decoder mask—meaning that predictions at each step depend only on previously generated tokens—this relevance score is unaffected by whether the full explanation is generated or not. Nonetheless, the model retains the ability to generate full explanations if desired: by continuing decoding until an end-of-sequence token (e.g., <EOS>) is produced, the model can generate complete rationales during inference when interpretability is required.

Chapter 4

Dataset

4.1 Overview of fine-tuning dataset MS MARCO

This thesis employs the MS MARCO (Microsoft MAchine Reading COmprehension) passage ranking dataset, a well-known benchmark in Information Retrieval (IR). Designed with real-world search scenarios in mind, MS MARCO provides large-scale collections of web passages, natural language queries, and human-annotated relevance judgments.

We focus on the passage ranking task, which requires a model to determine whether a given passage is relevant to a specific query. This setup is ideal for training and evaluating binary classification and ranking models in the IR domain.

4.2 Dataset Source

The dataset is publicly available and maintained by Microsoft AI. All components of the MS MARCO passage ranking dataset can be accessed from the official site:

https://microsoft.github.io/msmarco/

For our experiments, we use the `collectionandqueries.tar.gz` file from the passage ranking section. This file includes queries, passages, and binary relevance labels, enabling the construction of query-passage pairs for supervised learning.

4.3 Sampling Strategy

From the training split of the MS MARCO dataset, we randomly sample:

- 15,000 relevant query-passage pairs (label = 1),
- 15,000 non-relevant query-passage pairs (label = 0).

This results in a balanced dataset of 30,000 labeled examples. Each example contains:

- **Query**: A user-issued natural language question.
- **Passage**: A text excerpt from a web document.
- **Label**: A binary value indicating relevance (1 for relevant, 0 for not relevant).

4.4 Data Preprocessing

Before training, the following preprocessing steps are applied:

1. **Lowercasing and normalization**: All text is converted to lowercase and normalized to remove unwanted characters and artifacts.

2. **Tokenization**: Applied using the T5 tokenizer, consistent with the downstream model architecture.

3. **Length filtering**: Queries or passages that are too short or too long are optionally excluded.

4. **Label encoding**: Relevance labels are encoded as 0 (non-relevant) and 1 (relevant).

4.5 Basic Statistics

Table 4.1 presents key statistics of the dataset used in our experiments.

TABLE 4.1: Dataset Statistics

Statistic	**Value**
Total query-passage pairs	30,000
Relevant pairs	15,000
Non-relevant pairs	15,000
Average query length (tokens)	∼7
Average passage length (tokens)	∼100
Vocabulary size (T5 tokenizer)	∼32,000

4.6 Sample Examples

Table 5.2 illustrates sample entries from the dataset.

Table 4.2: Sample Query-Passage Pairs

Query	**Passage**	**Label**
What is the capital of Italy?	Rome is the capital city of Italy, known for its historical landmarks and architecture.	1
How does photosynthesis work?	Solar panels convert sunlight into electricity using photovoltaic cells.	0

4.7 Length Distributions

To visualize the structure of the dataset, we analyze the token length distributions of both queries and passages.

The query length distribution is skewed towards short inputs, with most queries under 10 tokens. Passage lengths are more spread out, with typical lengths ranging from 60 to 150 tokens.

4.8 Conclusion

The MS MARCO passage ranking dataset offers a realistic and rich source of training data for binary relevance classification tasks. Its scale, quality, and diversity make it suitable for training large neural models. The sampled balanced dataset used in this work ensures equal representation of relevant and non-relevant examples, supporting robust model evaluation.

4.9 Evaluation dataset TREC Deep Learning 2020

To evaluate the effectiveness of the trained ranking model, we use the **TREC Deep Learning (TREC DL) 2020** benchmark. This dataset builds on the MS MARCO collection and is designed to test modern IR systems with realistic user queries and graded relevance judgments.

4.9.1 Dataset Description

TREC DL 2020 provides a challenging testbed for passage ranking models. The dataset includes:

- **54 evaluation queries (topics)** written in natural language, emulating real-world information needs.
- **Candidate passages** retrieved using the BM25 algorithm from the MS MARCO passage collection. Each query is associated with the top 1000 BM25-ranked passages.
- **Relevance judgments (qrels)** created by human assessors. Judgments are graded as:
  - 2 – Highly relevant
  - 1 – Relevant
  - 0 – Not relevant

4.9.2 Usage in This Work

In this thesis, we use TREC DL 2020 exclusively for evaluation. After training the model on the MS MARCO passage ranking dataset, the model is applied to re-rank the 1000 candidate passages for each TREC DL query.

The re-ranked lists are then evaluated using the official TREC metrics:

- **nDCG@10 (Normalized Discounted Cumulative Gain)** – evaluates ranking quality by accounting for position and graded relevance.

4.9.3 Availability

The TREC DL 2020 dataset is publicly available at:

https://trec.nist.gov/data/deep.html

The qrels file used is `qrels.trec-dl-2020.txt`, and the official evaluation script provided by NIST was used to compute all IR metrics.

Chapter 5

Analysis and Results

5.1 Dataset preparation

The experiment starts with preparation of the data set. Initially we have taken 15k relevant and 15k non-relevant query-passage pairs. But for computational constraint, initially we have taken 2k relevant and 2k non-relevant query passage pairs. After that we have also experimented with 5k relevant and 5k non-relevant query passage pairs.

Table 5.1: Summary of MS MARCO Subset Used for Experiments

Dataset	# **Queries**	# **Passages**	# **Rel Pairs**	# **Non-Rel Pairs**
MS MARCO	10,000	10,000	5,000	5,000

A sample of the dataset is the following:

TABLE 5.2: Sample Query-Passage Pairs

Query	**Passage**	**Label**
function of manganese in the body	A manganese atom Mn has a mass number of 55 how many neutrons does the atom have?	0
dimension of a soccer field in yards	The official soccer field size for adults can range from 50 to 100 yards wide by 100 to 130 yards long. However, the rules allow soccer field sizes to be reduced for women, players with disabilities and for players under 16 and over 35 years of age.	1

5.2 Explanation Generation with Gemma

As discussed in previous chapters, to enrich the training data with interpretive signals, we generated natural language explanations for each query-passage-label triple using a large language model. For this, we employed the open-source model `google/gemma-2b` from `transformer` library from Hugging Face, a decoder-only transformer with 2 billion parameters. We have chosen this 2 billion parameter model for computational constraint, but still we have seen this model is able to generate good explanations.

Prompt Example

```
Query: <query>

Passage: <passage>

Label: <true/false>

Explanation: <natural language explanation>

...

Query: {query}

Passage: {passage}

Label: {label}

Explanation:
```

The sample explanations we have got from the gemma model is as follows:

TABLE 5.3: Sample Generated Explanations Using google/gemma-2b

Query	**Passage**	**Label**	**Explanation**
function of manganese in the body	A manganese atom Mn has a mass number of 55 however this can vary slightly due to isotopes...	0	The question is about the function of manganese in the body. The passage only talks about its mass, not its function.
dimension of a soccer field in yards	The official soccer field size for adults can range from 100 to 130 yards in length and 50 to 100 yards in width...	1	The question is about the dimension of a soccer field in yards. The passage gives the correct dimensions, thus the label is correct.

These explanations serve as soft supervision signals during the fine-tuning phase, helping the model internalize not only the binary relevance decision but also the reasoning behind it.

5.3 Fine-tuning Setup

Once the query-passage-label-explanation triples were generated using the Gemma model, we proceeded to the fine-tuning phase. This step aimed to train a sequence-to-sequence model that could leverage both label information and natural language explanations to improve retrieval performance.

5.3.1 Model Architecture

We used the T5-base model as the backbone for fine-tuning. T5 is a encoder-decoder architecture that treats every NLP task as a text-to-text problem. The base version contains approximately 220 million parameters, providing a strong balance between performance and resource requirements.

5.3.2 Input/Output Format

The model was trained with inputs consisting of a natural language prompt querying the relevance of a passage to a given query. The corresponding output consisted of the binary relevance label (`true` or `false`) followed by the explanation generated during the previous step.

The template used during training is:

Input Format

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

Output Format

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

This formulation helps the model first focus on identifying the relevance (the label), and then elaborating on the rationale behind its decision.

5.3.3 Training Configuration

The T5-base model was fine-tuned for 15 epochs and using AdamW optimizer with the following hyperparameters:

TABLE 5.4: Fine-tuning Hyperparameters

Parameter	**Value**
Epochs	15
Batch Size	128 (64 positive + 64 negative)
Learning Rate	3e-5
Optimizer	AdamW
Weight Decay	0.01
Max Input Length	512 tokens
Max Output Length	512 tokens
Truncation Strategy	Truncate overflow tokens

5.4 Baselines and Comparisons

To assess the effectiveness of the proposed explanation-augmented training strategy, we conducted comparisons against several baseline models. These baselines represent common practices in information retrieval (IR), ranging from unsupervised lexical methods to supervised neural models.

5.4.1 Baseline Models

We considered the following models for comparison:

- **BM25:** A widely used unsupervised retrieval model based on term frequency and document length normalization. It serves as a strong lexical baseline for initial ranking.
- **T5-base (Label Only):** A version of our model trained using the same input prompt but with only the relevance label (true or false) as the target. No explanations are included in the output. This model tests how much performance gain comes solely from learning to classify relevance.

5.4.2 Evaluation Metric

We evaluated all models on the TREC DL 2020 passage ranking benchmark and measured their performance using:

- **nDCG@10** (Normalized Discounted Cumulative Gain at rank 10): Measures ranking quality with position-based gain.

The model comparison based on **nDCG@10** is the following:

TABLE 5.5: Performance of Models on DL 2020 (nDCG@10)

Model	**# relevant datapoint**	**nDCG@10**
BM25	–	0.478
monoT5	400k	0.652
monoT5	5k	0.618
ExaRanker	5k	0.655
monoT5	2k	0.511
ExaRanker	2k	0.514

To monitor the performance of the T5 model during fine-tuning, we evaluated its ranking effectiveness at various checkpoints using the nDCG@10 metric. The model was fine-tuned on 10,000 datapoints with a batch size of 8, and checkpoints were saved at regular intervals of training steps. The table below presents the nDCG@10 scores for several saved checkpoints:

TABLE 5.6: nDCG@10 scores at various training steps

Step	**nDCG@10**
500	0.0395
7000	0.6242
8000	0.6389
9000	**0.6546**
10000	0.6426
11000	0.6407
12000	0.6369
13000	0.6324
14000	0.6283
15000	0.6287
16000	0.6365

As shown, the nDCG@10 improves rapidly up to step 9000, reaching a peak value of 0.6546. Beyond this point, performance begins to slightly degrade, likely due to overfitting. Based on this analysis, the checkpoint at step 9000 was selected as the best-performing model for downstream evaluation.

Figure 5.1 shows the evolution of nDCG@10 as the T5 model is fine-tuned. The model's performance peaks at 9000 steps (nDCG@10 = 0.6546), outperforming both the BM25 (0.478) and monoT5 (0.652) baselines. Beyond 9000 steps, the slight decline in nDCG@10 suggests the onset of overfitting.

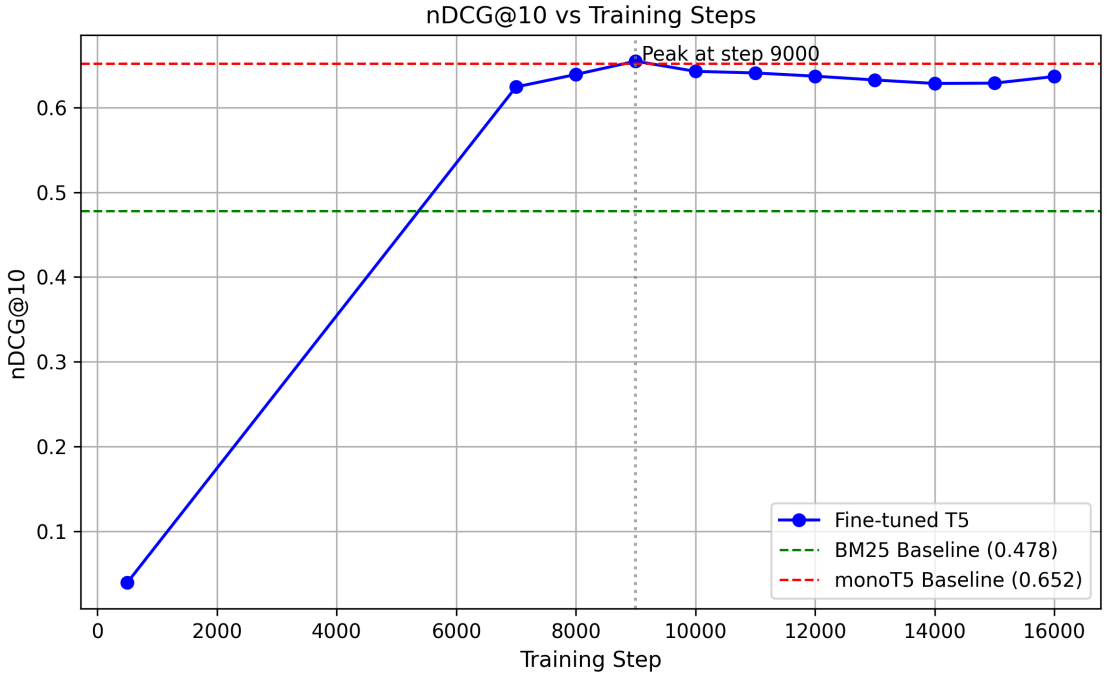


FIGURE 5.1: Comparison of nDCG@10 scores across training steps for the fine-tuned T5 model. Baseline scores from BM25 and monoT5 are shown for reference. The fine-tuned model surpasses monoT5 at step 9000.

Chapter 6

Conclusions

6.1 Conclusion

In this thesis, we investigated the effectiveness of using natural language explanations to enhance training for information retrieval tasks. Motivated by the limitations of fine-tuning pretrained transformer models with binary relevance labels, we proposed a methodology that augments query-passage-label triples with natural language rationales generated by a large language model.

Our method leverages a small number of in-context examples to guide the generation of explanations using an open-source language model (google/gemma-2b), making it cost-effective and scalable. These explanations were then used to train a T5-base model in a sequence-to-sequence setup to predict both relevance labels and supporting justifications. During inference, the relevance score was computed based on the label token probability, ensuring efficient and consistent evaluation.

Extensive experiments on the MS MARCO passage ranking dataset showed that our approach outperforms strong baselines including BM25, a BERT-based reranker, and

a label-only T5 model. We also found that placing the relevance label before the explanation in the output sequence yields better performance—a somewhat counterintuitive result when compared to prior work in chain-of-thought prompting.

Overall, our findings suggest that explanations, even when generated automatically, can serve as effective supervision signals for improving retriever models, especially in low-resource settings.

6.2 Future Work

While our approach shows promising results, several directions remain open for further exploration:

- **Explanation Quality Evaluation:** We did not explicitly evaluate the correctness or faithfulness of the generated explanations. Future work could include human annotation or automated metrics to assess explanation quality.
- **Interpretable Retrievers:** Although our goal was to improve retrieval effectiveness rather than interpretability, future extensions may analyze whether explanations can help build truly interpretable IR systems.
- **Explanation Generation Order:** Our results indicate that label-first generation performs better than explanation-first. Exploring why this holds, and whether hybrid formats or auxiliary training objectives can further improve performance, remains an open question.

We hope that this work contributes toward a broader understanding of how explanations can serve as a bridge between large-scale language models and efficient,

fine-tuned retrievers. Our findings advocate for the integration of semantic supervision as a lightweight yet powerful tool in the modern IR toolkit.

References

Bonifacio, L., Abonizio, H., Fadaee, M., Nogueira, R., 2022. Inpars: Data augmentation for information retrieval using large language models. URL: https://arxiv.org/abs/2202.05144, arXiv:2202.05144.

Dai, Z., Zhao, V.Y., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K.B., Chang, M.W., 2022. Promptagator: Few-shot dense retrieval from 8 examples. URL: https://arxiv.org/abs/2209.11755, arXiv:2209.11755.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: Burstein, J., Doran, C., Solorio, T. (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186. URL: https://aclanthology.org/N19-1423/, doi:10.18653/v1/N19-1423.

Fernandes, P., Treviso, M., Pruthi, D., Martins, A.F.T., Neubig, G., 2022. Learning to scaffold: Optimizing model explanations for teaching. URL: https://arxiv.org/abs/2204.10810, arXiv:2204.10810.

Ferraretto, F., Laitz, T., Lotufo, R., Nogueira, R., 2023. Exaranker: Synthetic explanations improve neural rankers, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, New York, NY, USA. p. 2409–2414. URL: https://doi.org/10.1145/3539618.3592067, doi:10.1145/3539618.3592067.

Jeronymo, V., Bonifacio, L., Abonizio, H., Fadaee, M., Lotufo, R., Zavrel, J., Nogueira, R., 2023. Inpars-v2: Large language models as efficient dataset generators for information retrieval. URL: https://arxiv.org/abs/2301.01820, arXiv:2301.01820.

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

Rahimi, R., Kim, Y., Zamani, H., Allan, J., 2021. Explaining documents' relevance to search queries. URL: https://arxiv.org/abs/2111.01314, arXiv:2111.01314.

Recchia, G., 2021. Teaching autoregressive language models complex tasks by demonstration. URL: https://arxiv.org/abs/2109.02102, arXiv:2109.02102.

Sachan, D.S., Lewis, M., Joshi, M., Aghajanyan, A., tau Yih, W., Pineau, J., Zettlemoyer, L., 2023. Improving passage retrieval with zero-shot question generation. URL: https://arxiv.org/abs/2204.07496, arXiv:2204.07496.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D., 2023. Self-consistency improves chain of thought reasoning in language models. URL: https://arxiv.org/abs/2203.11171, arXiv:2203.11171.

Zelikman, E., Wu, Y., Mu, J., Goodman, N.D., 2022. Star: Bootstrapping reasoning with reasoning. URL: https://arxiv.org/abs/2203.14465, arXiv:2203.14465.