



University of  
Massachusetts  
Amherst

## Leveraging Explanations for Information Retrieval Systems under Data Scarcity

Item Type	Dissertation (Open Access)
Authors	Yu, Puxuan
DOI	<a href="https://doi.org/10.7275/55184">10.7275/55184</a>
Rights	Attribution 4.0 International
Download date	2026-03-06 09:13:55
Item License	<a href="http://creativecommons.org/licenses/by/4.0/">http://creativecommons.org/licenses/by/4.0/</a>
Link to Item	<a href="https://hdl.handle.net/20.500.14394/55184">https://hdl.handle.net/20.500.14394/55184</a>

**LEVERAGING EXPLANATIONS FOR INFORMATION  
RETRIEVAL SYSTEMS UNDER DATA SCARCITY**

A Dissertation Presented

by

PUXUAN YU

Submitted to the Graduate School of the  
University of Massachusetts Amherst in partial fulfillment  
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2024

Manning College of Information and Computer Sciences

© Copyright by Puxuan Yu 2024

All Rights Reserved

# LEVERAGING EXPLANATIONS FOR INFORMATION RETRIEVAL SYSTEMS UNDER DATA SCARCITY

A Dissertation Presented

by

PUXUAN YU

Approved as to style and content by:

---

James Allan, Co-chair

---

Razieh Rahimi, Co-chair

---

Hamed Zamani, Member

---

Qingyao Ai, Outside Member

---

Ramesh K. Sitaraman, Associate Dean for  
Educational Programs and Teaching  
Manning College of Information and Computer  
Sciences

## ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, James Allan. Our first meeting was at SIGIR 2018, when I had just been admitted to UMass without a designated advisor. James kindly welcomed me into the lab without knowing much about me and has provided me with immense flexibility to explore my research interests within IR since then. He has consistently supported my research ideas and helped me see the bigger picture when I was bogged down in details. His wisdom, patience, and sense of humor have made my PhD journey both fulfilling and enjoyable, strengthening my commitment to IR research for the future.

I would also like to extend my heartfelt thanks to my committee members, Negin Rahimi, Hamed Zamani, and Qingyao Ai. I have had the privilege of working closely with Negin throughout my PhD, benefiting immensely from her help in brainstorming ideas, refining my writing style, and solving complex problems together, often through drawing elaborate flowcharts on paper. I will always cherish the late-night paper-writing frenzies we shared as deadlines loomed. I have been fortunate to briefly be labmates with Hamed and Qingyao and then witness their rise to superstars in IR research. They are all my role models, and I am honored and grateful to have had them review and provide feedback on my work.

I would like to thank the CIIR members for their unwavering support over the years. My appreciation extends to staff members Jean Joyce, Dan Parker, Kate Moruzzi, Michael Schwendenmann, Michael Zarozinski, and Gregory Brooks. Additionally, I am grateful to my lab mates, in no particular order: Keping, Liu, Youngwoo, Yen-Chieh, Ali, Shahrzad, Myung-ha, Chen, Rab, Sheikh, Lakshmi, Tanya, Nazanin, Alireza, Chris, Hansi, and Yaxin. A special thanks to Zhiqi for being not

just a lab mate and collaborator, but also a great friend and big brother. I have learned so much from my interactions with each of you.

I am also grateful to my mentors outside of UMass. I thank Professor Hongning Wang for introducing me to IR research during my undergraduate visit to the University of Virginia. I also thank my internship mentors: Hongliang Fei and Ping Li at Baidu Research USA, Antonio Mallia and Matthias Petri at Amazon Alexa, and Daniel Cohen and Hemank Lamba at Dataminr. They have been incredibly supportive of my pursuit of my own research agenda in various environments and guided me in choosing industry research as the next step in my career.

I owe a special thanks to my family for their emotional support during the five years I was unable to visit home. I am particularly grateful to my parents for instilling in me the core values of honesty and humility, which I hold dear in both research and life. Their unwavering support of my education has been instrumental in reaching this stage of academic achievement.

Finally, I want to express my deepest appreciation to my wife, Danni. I never imagined I would get married during my PhD journey, but it has turned out to be my greatest achievement during this time. Her love and support have made me a better person and helped me overcome difficult times.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-1617408, in part by NSF grant #IIS-2039449, in part by NSF grant #2106282, in part by the AFRL and IARPA contract #FA8650-17-C-9118 under Raytheon BBN Technologies Corporation subcontract #14775, and in part by Amazon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

## ABSTRACT

# LEVERAGING EXPLANATIONS FOR INFORMATION RETRIEVAL SYSTEMS UNDER DATA SCARCITY

SEPTEMBER 2024

PUXUAN YU

B.Eng., WUHAN UNIVERSITY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan and Professor Razieh Rahimi

The importance of explanations in the advance of information retrieval (IR) systems is on the rise. On one hand, this is driven by the increasing complexity of IR systems and the demand for transparency and interpretability from users; on the other hand, explanations can inherently improve the effectiveness of IR systems without necessarily being displayed to users. However, the scarcity of data poses significant challenges in developing these explanations, as acquiring high-quality explanations for relevance judgments is prohibitively expensive yet crucial for training neural network-based IR models and explanation generation models. To overcome these challenges, we utilize open-domain knowledge and generative language models to facilitate the generation of user-oriented explanations for various IR tasks limited by data availability.

We start by introducing a novel model-agnostic task for search result explanations that emphasizes context-aware summaries, detailing each document’s relevance to the query and other documents. To address this task, we design a novel Transformer-based encoder-decoder architecture. Next, we develop an inherently explainable IR model specifically designed to provide diversified reranking of retrieved documents. This model is pre-trained on open-domain data using explanation tasks, achieving state-of-the-art results in search result diversification with minimal domain-specific data. Additionally, we explore how natural language explanations can enhance the capabilities of generative language models to augment IR datasets through synthetic query generation, achieved by automatically identifying similarities and differences between document pairs. Finally, we utilize zero-shot generative language models to directly elicit natural language explanations of relevance between search queries and candidate documents, providing crucial auxiliary information for the calibration of neural ranking models and thus enhancing their ability to generate meaningful scores.

# TABLE OF CONTENTS

	Page
<b>ACKNOWLEDGMENTS</b> .....	iv
<b>ABSTRACT</b> .....	vi
<b>LIST OF TABLES</b> .....	xiii
<b>LIST OF FIGURES</b> .....	xv
 <b>CHAPTER</b>	
<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1 Listwise Model-agnostic Explanation Generation for SERPs .....	4
1.2 Intrinsically Explainable End-to-end Search Result Diversification .....	6
1.3 Data Augmentation with Explanation-enhanced Large Language Models .....	9
1.4 Leveraging Zero-shot Explanations for IR Tasks with Scarce Data .....	11
1.5 Summary .....	13
<b>2. RELATED WORK</b> .....	<b>14</b>
2.1 Neural Ad-hoc Retrieval .....	14
2.2 Explainable Information Retrieval .....	17
2.2.1 Post-hoc Interpretability .....	17
2.2.2 Intrinsic Interpretability .....	19
2.3 Explanations and Generative Large Language Models .....	21
2.3.1 Explanations to LLMs .....	22
2.3.2 Explanations by LLMs .....	22
<b>3. LISTWISE MODEL-AGNOSTIC EXPLANATION     GENERATION FOR SERPS</b> .....	<b>24</b>

3.1	Multi-aspect and Listwise Search Result Explanation . . . . .	26
3.1.1	Wikipedia as a Weakly Labeled Dataset . . . . .	27
3.1.2	Adapting MIMICS Datasets for Evaluation . . . . .	29
3.2	Listwise Explanation Generator (LiEGe) . . . . .	30
3.2.1	Input Representation . . . . .	31
3.2.2	Encoder . . . . .	32
3.2.3	Decoder . . . . .	34
3.2.4	Training of LiEGe . . . . .	35
3.3	Experimental Setup . . . . .	36
3.3.1	Competing Methods . . . . .	36
3.3.2	Evaluation Metrics . . . . .	37
3.4	Experimental Results and Analysis . . . . .	38
3.4.1	Comprehensive Explanation Generation on Wiki . . . . .	39
3.4.2	Novelty Explanation Generation on Wiki . . . . .	40
3.4.3	Explanation Generation on MIMICS . . . . .	40
3.4.4	Ablation Study . . . . .	42
3.5	Summary . . . . .	44
<b>4.</b>	<b>INTRINSICALLY EXPLAINABLE END-TO-END SEARCH</b>	
	<b>RESULT DIVERSIFICATION . . . . .</b>	<b>45</b>
4.1	Diversification Using Bottlenecks (DUB) . . . . .	47
4.1.1	Task Formulation and Model Overview . . . . .	47
4.1.2	Text Encoder . . . . .	48
4.1.3	Neural Aspect Extractor . . . . .	50
4.1.3.1	Aspect Extractor Using Multi-Head Attention . . . . .	50
4.1.3.2	Aspect Extractor Using GDKM-based Clustering . . . . .	51
4.1.4	Diversified Ranker . . . . .	54
4.2	Addressing Data Scarcity with Explanation-based Pre-training . . . . .	55
4.2.1	Pre-training with Aspect Matching . . . . .	55
4.2.1.1	Aspect Pre-training Data . . . . .	56
4.2.1.2	Aspect Matching Task . . . . .	56
4.2.1.3	Optimal-Transport Based Objective . . . . .	56

4.2.1.4	Teacher-Forcing Based Objective .....	57
4.2.2	End-to-end SRD Training .....	57
4.3	Experimental Setup .....	58
4.3.1	Evaluation Datasets .....	58
4.3.2	Competing Methods .....	59
4.3.3	Evaluation Metrics .....	60
4.4	Experimental Results and Analysis .....	60
4.4.1	Importance of Supervised Aspect Extraction .....	60
4.4.2	Utility of Pre-training .....	61
4.4.3	Diversification on MIMICS-Div .....	62
4.4.4	Discussion: MHA vs. GDKM .....	62
4.5	Evaluation of Latent Aspects .....	63
4.5.1	Compared Methods .....	63
4.5.2	Measuring Diversity .....	64
4.5.3	Measuring Relevance .....	65
4.6	Summary .....	66
<b>5.</b>	<b>EXPLANATION-GUIDED DATA AUGMENTATION USING GENERATIVE LANGUAGE MODELS .....</b>	<b>67</b>
5.1	Contrastive Query Generation (CQG) .....	69
5.1.1	Task Definition .....	69
5.1.2	Contrasting Documents Mining .....	70
5.1.3	Query Generation with Explanation-Guided Prompting .....	71
5.1.4	Data Cleaning and Verification .....	73
5.1.4.1	Format-based Filtering .....	73
5.1.4.2	Answerability Verification with Self Reflection .....	74
5.1.4.3	Consistency Filtering .....	74
5.2	Experimental Setup .....	75
5.2.1	Evaluation Setup .....	75
5.2.2	Datasets .....	75
5.2.3	Competing Methods .....	76
5.3	Experimental Results and Analysis .....	77

5.3.1	Effectiveness of Contrastive Query Generation	77
5.3.2	Ablation Studies	78
5.3.2.1	Number of Samples from Consistency Filtering	78
5.3.2.2	Effect of Entity-based Filtering	79
5.3.3	Error Analysis	80
5.4	Summary	82
<b>6.</b>	<b>AUTO-GENERATED EXPLANATION OF RELEVANCE FOR SCALE CALIBRATION</b>	<b>83</b>
6.1	Scale Calibration of Neural Ranking Models	86
6.2	Scale Calibration with Natural Language Explanations	87
6.2.1	Overview	87
6.2.2	Acquiring Natural Language Explanations via LLM Prompting	88
6.2.2.1	Literal Explanation	89
6.2.2.2	Conditional Explanation	90
6.2.3	Aggregating Multiple Monte Carlo Explanations	90
6.3	Experimental Setup	92
6.3.1	Data	93
6.3.2	Metrics	93
6.3.3	Competing Methods	96
6.3.4	Implementation Details	99
6.4	Experimental Results and Analysis	100
6.4.1	Utilities of Natural Language Explanations	100
6.4.2	Consistency across Different Training Objectives	101
6.4.3	Ablations on Natural Language Explanations	102
6.5	Summary	104
<b>7.</b>	<b>CONCLUSIONS AND FUTURE WORKS</b>	<b>105</b>
7.1	Conclusions	105
7.2	Future Works	107
7.2.1	Expanding the Scope of Context-Aware Explanations	107
7.2.2	Incorporating User Feedback	108

7.2.3 Improving Factuality and Faithfulness of Automatically  
Generated Explanations .....109

**BIBLIOGRAPHY ..... 111**

## LIST OF TABLES

Table	Page
3.1	Statistics of created datasets for the new SeRE task. . . . . 28
3.2	Results for comprehensive explanation generation on the Wiki-SA dataset (single-aspect). . . . . 38
3.3	Results for comprehensive explanation generation on the Wiki-CEG dataset (multi-aspect). . . . . 38
3.4	Results for novelty explanation generation evaluated on Wiki-NEG. . . . . 40
3.5	Results for CEG evaluated on MIMICS-CEG. . . . . 41
3.6	Results for NEG evaluated on MIMICS-NEG. . . . . 41
3.7	Performance of ablated variants of the LiEGe model. Symbol $\nabla$ shows statistical significant differences comparing with LiEGe. . . . . 43
4.1	Search result diversification on TREC-Web. . . . . 61
4.2	Search result diversification on MIMICS-Div. . . . . 62
4.3	Evaluating the quality of extracted aspects. . . . . 64
5.1	Statistic of evaluation datasets used in the experiments. . . . . 76
5.2	Evaluation results of baseline methods and CQG on five query sets. “nDCG” represents nDCG@10. Statistical significance (t-tests with Bonferroni correction at the 95%) over the strongest baseline is marked with $\dagger$ . . . . . 77
6.1	Statistics of the TREC-DL 2019-2022 and NTCIR-14 WWW-2 Datasets. The lengths of queries and documents are quantified using BERT tokenization. For the NTCIR dataset, documents sourced from ClueWeb have undergone preprocessing to retain only the initial 512 tokens. . . . . 94

6.2	Ranking and scale calibration performance of baseline methods and our approaches on the TREC dataset. Statistically significant improvements over “Platt Scaling monoBERT” are marked with †.....	100
6.3	Ranking and scale calibration performance of baseline methods and our approaches on the NTCIR dataset. Statistically significant improvements over “Platt Scaling monoBERT” are marked with †.....	101
6.4	The effect of different types of natural language explanations and selection strategies on the ranking and scale calibration performance of neural rankers. ....	103

## LIST OF FIGURES

Figure	Page
3.1 Explaining a search engine result page (SERP) in different settings. . . . .	25
3.2 An example of creating multi-aspect documents from the English Wikipedia for the CEG and NEG task. . . . .	27
3.3 The architecture of LiEGe. Panel (b) presents a high-level overview of the encoder-decoder structure, with specific modifications for listwise explanation generation highlighted in red. Detailed descriptions of these changes are provided in panels (a) and (c), respectively. . . . .	31
4.1 An overview of the DUB framework. . . . .	48
4.2 Aspect extraction using GDKM clustering ( $K=3, \nu=2$ ). Each cluster contains passages that are inside its border and passages that point to its border. E.g., both $p_0$ and $p_4$ are part of the yellow cluster, with the former having a higher probability, whereas $p_3$ does not belong to this cluster. . . . .	53
5.1 Performance of finetuned RankT5 with varying numbers of top consistency filtered synthetic data, evaluated on in-domain TREC query sets. . . . .	79
5.2 Performance of finetuned RankT5 with varying numbers of top consistency filtered synthetic data, evaluated on out-of-domain datasets FiQA and NQ. . . . .	80
6.1 The key idea of leveraging natural language explanations for scale calibration: Neural ranking models struggle to produce meaningful ranking scores when encountering complex query-document pairs. We investigate the integration of natural language explanations as inputs to neural rankers, aiming to simplify the scale-calibrated ranking task for these rankers. . . . .	85

6.2	Reliability diagrams for two models on TREC: The left diagram shows a model with ranking scores densely concentrated on the lower part of the scale, which exhibits better ECE performance due to ECE’s failure to account for prediction coverage across the target scale. On the right, the CB-ECE penalizes this undesirable behavior, indicating that the model providing better coverage across the scale is more effectively calibrated.....	95
6.3	Ranking and scale calibration performance of the baseline (neural ranker taking query and documents) and NLE-based approaches on TREC, using four different optimization objectives. NLE-based approaches consistently yield better ranking (left) and calibration (right) performance.....	102

# CHAPTER 1

## INTRODUCTION

The concept of explanation in information retrieval (IR) is flexible and multifaceted, encompassing any element that enhances the clarity of an IR system. These explanations can take various forms, such as snippets, highlighted terms or sentences, natural language summaries, and simplified approximations of complex neural IR models. Additionally, explanations in IR can be customized for diverse audiences, including IR specialists, lawyers, regulators, regular search engine users, or even the IR systems themselves. Thus, the relationship between explanations and IR systems is complex.

The role of explanations is increasingly crucial in advancing IR systems for several reasons. As search engines and recommendation systems become deeply integrated into user-centric technologies, users face unprecedented information overload (Montebello, 1998). Furthermore, as the algorithms and models powering these systems grow more sophisticated, there is a rising skepticism about their reliability, as people often distrust what they cannot understand. In this context, explanations not only enhance user experience by helping users navigate vast amounts of information more efficiently, but also foster trust by improving their understanding of the systems. Moreover, as artificial intelligence (AI) models begin to exhibit human-like cognitive traits, the concept of explanations extends beyond human audiences, also serving to refine machine learning models and, consequently, the IR systems themselves.

Following our analysis of the roles explanations play in IR systems, we identify two primary use cases: improving interpretability and enhancing effectiveness. To improve

interpretability, we aim to generate accurate and effective explanations. Conversely, to enhance effectiveness, our goal is to refine IR systems—primarily by delivering more relevant results—with the aid of explanations during key phases of model development, such as training and inference. This dissertation examines both areas, specifically targeting the needs of regular search engine users and machine-based applications. Since an enhanced search engine benefits all users, our overarching aim in both scenarios is to enrich the user experience with the search engine.

Data scarcity poses a significant challenge for studying explanations and text search systems. When aiming to enhance the interpretability of text search systems by generating explanations, “gold” explanations—those that articulate the relevance between a search query and a candidate document—become crucial for training and evaluating such models. However, annotators expend considerably more effort providing these detailed justifications compared to simply rendering judgments, making the acquisition of high-quality explanation data prohibitively expensive. Additionally, in efforts to improve search systems’ ranking capabilities, the underlying neural ranking models require substantial amounts of data to perform effectively. Consequently, whether explanations are involved or not, neural ranking models consistently struggle with the issue of data scarcity.

In terms of improving the interpretability of IR systems, providing concise and natural language explanations on search engine result pages (SERPs) has proven to be an easily understandable and widely accepted approach for normal search engine users (Iwata et al., 2012a; Rahimi et al., 2021). This strategy focuses on explaining *the outcomes* of IR models, making it model-agnostic and highly adaptable. Currently, a significant challenge with the organization of search results in modern search engines, along with common explanation methods like keyphrase extraction (Zhang et al., 2019a) and snippet generation (Chen et al., 2020b), is the insufficient differentiation of relevant documents. For example, consider the titles “UMass Amherst - Campus

- University of Massachusetts” and “University of Massachusetts Amherst - Niche” on the first SERP for the query “UMass Amherst” from Google Search. The titles and snippets alone do not clearly distinguish between these two documents. Users on information-seeking tasks must thoroughly examine both documents to determine if their information needs are met. Instead, we propose *explanations in context* that elucidate the relations and differences between documents on the SERP, enabling users to easily identify relevant information and take subsequent actions. For instance, by using explanations in context, we can specify that the first document discusses the “inclusive culture” at UMass Amherst, while the second does not, thereby saving users from the need to extensively review both documents.

Additionally, offering more nuanced subtopics as explanations for under-specified queries is closely linked to search result diversification, which aims to improve the fairness of exposure of multiple subtopics in the final rankings. By first identifying relevant subtopics in each of the top-ranked documents, an algorithm can then re-rank these documents to enhance diversity. To simultaneously provide explanations and achieve diversified ranking based on the explanations, the diversification model should be end-to-end learnable, ensuring that interpretability is maintained without compromising the effectiveness of the rankings. For effective end-to-end learning, explanations must be grounded in the input queries and documents, and integrated as an intrinsic component of the diversification model.

The two aforementioned tasks—generating explanations in context and integrating explanation with diversification—both confront the issue of data scarcity. Ideally, training and evaluating such models would require expert annotators to identify relevant subtopics of the original search query and map these to top-ranked documents. However, this process is prohibitively expensive in practice. To address the data scarcity problem, we exploit the similarity between the relation of section text to headings in structured content on Wikipedia and the relation of document text to

explanations in text search engines. We have found that models can be pre-trained using large-scale self-supervised data from open domains to achieve satisfactory performance in real web search settings. These models can then be further fine-tuned with minimal web search data to enhance their performance, demonstrating significant transfer learning capabilities from resource-rich to resource-lean domains.

To address the data scarcity challenge in developing more effective text ranking models, we combine natural language explanations with large language models (LLMs). Recent advances have made automated language generation by LLMs more accessible and cost-effective. We explore two applications of LLMs: offline data augmentation using explanations, and online elicitation of reasoning and uncertainty through explanations. In the first case, we focus on synthetic query generation, a prevalent method for data augmentation in IR. We leverage the in-context learning capabilities of LLMs to prompt the model to generate both explanations and synthetic queries, forming more useful training data from pairs of source and contrasting documents. In the second case, we address scale calibration, which requires that ranking scores from neural ranking models be meaningful while facilitating effective rankings. Due to the scarcity of training data labeled on the desired scale compared to the size of neural ranking models, we use zero-shot LLMs to generate natural language explanations as a medium for conveying reasoning and uncertainty. These explanations serve as inputs to the neural ranker, significantly easing the training burden and addressing the data scarcity issue.

## 1.1 Listwise Model-agnostic Explanation Generation for SERPs

We introduce a novel task named “Search Result Explanation” (SeRE) that focuses on generating multi-aspect summaries as explanations for all documents within search engine result pages (SERPs) jointly. This task differs from previous research (Rahimi et al., 2021), which primarily focused on providing individual ex-

planations for documents, with each explanation focusing on a single query aspect. The SeRE task consists of two sub-tasks: novelty explanation generation (NEG) and comprehensive explanation generation (CEG). The difference is that, in the NEG task, explanations for lower-ranked documents are required to avoid repetition of query aspects already covered by higher-ranked documents. To train and evaluate the SeRE task, we utilize the English Wikipedia to construct appropriate datasets.

To address this new task, we introduce modifications to the standard Transformer-based encoder-decoder architecture. Our model, referred to as Listwise Explanation Generator (LiEGe), incorporates a combination of local and newly proposed global layers in the encoder. Furthermore, we enhance the decoder by introducing a cross-document attention layer positioned between the original self-attention and encoder-decoder attention layers. This adaptation enables the encoder-decoder model to generate a single explanation per document, effectively utilizing other documents as contextual information.

Our experimental results demonstrate the effectiveness of LiEGe in delivering explanations for SERPs in both comprehensive explanation generation (CEG) and novelty explanation generation (NEG) scenarios. Notably, LiEGe exhibits significant improvements over state-of-the-art text generation techniques, with more notable gains observed in the NEG sub-task. This outcome underscores the importance of leveraging other documents as contextual information for NEG. In addition to utilizing automatically generated datasets from Wikipedia, we evaluate the performance of LiEGe on MIMICS (Zamani et al., 2020), a more realistic Web Search dataset derived from the Bing search engine. LiEGe proves to be effective in this real-world dataset as well. Moreover, we observe that pre-training LiEGe on Wikipedia yields additional enhancements in performance when applied to the MIMICS dataset.

The main contributions in this area include:

- *Contribution 1.1.* We establish a SERP explanation task (SeRE), wherein an explanation is capable of encompassing multiple aspects and can be influenced by the content of other documents within the SERP. We put forth two plausible sub-tasks: comprehensive explanation generation (CEG) and novelty explanation generation (NEG).
- *Contribution 1.2.* In order to facilitate the training and evaluation of models for these sub-tasks, we develop techniques for constructing appropriate datasets utilizing the English Wikipedia as a valuable resource. The abundance of data available from Wikipedia proves advantageous for training Transformer-based text generation models. We also propose the adaptation of MIMICS (Zamani et al., 2020) as more realistic datasets for our proposed task.
- *Contribution 1.3.* We present a novel explanation generation model based on the Transformer architecture, named Listwise Explanation Generator (LiEGe), designed specifically for addressing the SeRE task. We conduct a comprehensive set of experiments across multiple settings. Our experimental results consistently demonstrate the effectiveness of LiEGe across both sub-tasks and all train-evaluate settings. Additionally, our findings indicate successful knowledge transfer from Wikipedia to MIMICS. Quantitatively, LiEGe surpasses the strongest baseline model, BART (Lewis et al., 2020) by 7.7% in the CEG sub-task and by 27.1% in the NEG sub-task in terms of BLEU on MIMICS.

## 1.2 Intrinsically Explainable End-to-end Search Result Diversification

Model-agnostic explanations for search engine result pages (SERPs), as discussed in Section 1.1, are independent of the underlying IR model’s decision-making processes. This independence between the ranking and explanation generation processes

naturally raises the question: Is it feasible to generate explanations and perform ranking simultaneously? Specifically, can an IR model generate aspect-like explanations and rank documents based, at least in part, on these explanations?

Search result diversification (Santos et al., 2015) is an established task that involves reranking documents to cover as many different query aspects as possible in the top positions. When query aspects are explicitly known, either provided or derived from external resources, these diversification models are intrinsically explainable; both the query aspects and the extent to which documents cover these aspects are transparent and interpretable. However, traditional coverage-based SRD models depend on external sources or systems, such as Google query suggestions, to obtain query aspects. This reliance presents two major challenges: the query aspects used for explanations are not sourced from the candidate documents themselves, which may reduce their relevance and informativeness; additionally, separating explanation generation from the ranking process can impair the performance of the ranking model.

We introduce a novel framework called Diversification Using Bottlenecks (DUB), which incorporates a neural aspect extractor component to address these issues. The DUB framework utilizes the principles of the Information Bottleneck method (Tishby et al., 2000) to extract latent query aspects directly from candidate documents, optimizing them as bottlenecks for diversified reranking. The differentiable nature of DUB’s aspect extractor facilitates end-to-end learning of the entire framework, including the text encoder and diversified ranker, using aspect-level relevance judgment labels. Furthermore, when necessary, the latent query aspects can be translated into natural language to provide explanations.

The scarcity of data, however, presents a significant challenge, as existing SRD datasets do not offer adequate support for training our proposed framework. To overcome this, we leverage the English Wikipedia to pre-train the more parameter-intensive components of DUB on a related explanation task called *aspect matching*.

By pre-training part of the framework on Wikipedia and subsequently fine-tuning it on dedicated SRD datasets, DUB achieves impressive diversification performance. Compared to novelty-based (non-coverage) SRD approaches, DUB offers inherent explainability. In comparison to other coverage-based approaches, DUB achieves a similar level of explainability while eliminating the reliance on external sources for providing query aspects, thus enhancing its utility and effectiveness.

The main contributions in this area include:

- *Contribution 2.1.* We introduce DUB, a novel end-to-end learnable search result diversification framework comprising a text encoder, an aspect extractor, and a diversified ranker. The aspect extractor processes passage representations of top-ranked documents to produce optimized aspect representations for the diversified ranking task.
- *Contribution 2.2.* We develop a unique training approach for DUB, beginning with the initial training of part of the framework on the English Wikipedia, followed by comprehensive end-to-end fine-tuning on dedicated SRD datasets. Experimental results show that DUB outperforms the most robust existing SRD approach by achieving a significant 4.3% improvement in  $\alpha$ -nDCG@20, while retaining its explainability advantage.
- *Contribution 2.3.* We conduct further analysis and reveal that the query aspects generated by DUB align more closely with human judgments compared to other methods that utilize external systems for automatic aspect generation. This underscores DUB’s superior effectiveness in diversification.

### 1.3 Data Augmentation with Explanation-enhanced Large Language Models

Besides enhancing the interpretability and trustworthiness of search results, as discussed in Sections 1.1 and 1.2, explanations can also help address the data scarcity problem in information retrieval in general. This is facilitated by the rapid development of large language models (LLMs). Recent studies have explored the use of LLMs in IR from various angles, including query generation (Dai et al., 2022; Jeronimo et al., 2023; Mayfield et al., 2023), document generation (Askari et al., 2023; Gao et al., 2023), relevance feedback (Mackie et al., 2023), providing relevance judgments (Faggioli et al., 2023a), and even direct reranking (Ma et al., 2023b). Despite these efforts, the use of explanations coupled with LLMs in IR applications remains largely unexplored. Recognizing this gap, we examine how explanations could mitigate data scarcity and enhance IR model effectiveness, focusing particularly on the task of synthetic query generation—the most prevalent method of data augmentation.

We identify several issues with current synthetic query generation approaches. Firstly, although query generation is an effective data augmentation technique for adapting IR models to new domains, the majority of synthetic queries generated by LLMs tend to be relatively easy and generic. This raises questions about the approach’s effectiveness and whether its utility could be improved by generating more challenging and diverse queries. Secondly, it is common practice to use documents randomly sampled from the top results of a retriever as negative training samples. However, there is no assurance that these samples are not false negatives (actually relevant) or easy negatives (not-at-all relevant). Given that neural IR models predominantly use contrastive objectives for training, we hypothesize that model performance could be enhanced by ensuring that negative training documents are truly hard negatives.

Following these observations, we introduce a methodology that leverages explanation-enhanced LLMs to generate more effective training data. We develop a new framework called contrastive query generation (CQG), which differs from previous methods. Instead of generating a query from a single document and then sampling negative documents (generate-then-sample), our approach begins by sampling pairs of similar yet distinct documents and then generates a query based on these pairs (sample-then-generate). We employ explanation-infused prompts to guide LLMs in identifying similarities and discrepancies between the document pairs, which then inform the query generation process. Since the queries are derived from both the similarities and discrepancies, the contrasting documents used in their generation naturally constitute hard negatives. Consequently, this approach produces higher quality training data and results in better-trained IR models.

The main contributions in this area include:

- *Contribution 3.1.* We introduce the contrastive query generation (CQG) method for both in-domain and out-of-domain data augmentation tasks. This method leverages explanations to guide large language models in generating queries that naturally produce more useful IR training data, featuring hard negative samples.
- *Contribution 3.2.* To address the limitations of LLMs in this task, particularly in terms of instruction following and hallucination, we implement verification and filtering conditions. These automatically select superior training data from the generated synthetic outputs, resulting in significant improvements in the downstream IR models being trained.
- *Contribution 3.3.* We conduct extensive experiments to evaluate the performance of CQG on both in-domain and out-of-domain datasets. Notably, CQG surpasses the previous state-of-the-art open-source query generation approach (Jeronymo

et al., 2023) by 6.8% on DL-Hard (Mackie et al., 2021) and 7.1% on Natural Questions (Kwiatkowski et al., 2019), when used to finetune large neural rankers, as evaluated by nDCG@10. Additionally, we analyze the inherent limitations of LLMs performing the CQG task, highlighting the potential for improvement when these limitations are addressed.

## 1.4 Leveraging Zero-shot Explanations for IR Tasks with Scarce Data

In addressing the challenge of data scarcity in retrieval and ranking, particularly in domains lacking relevance data, the common strategy is data augmentation, as discussed in Section 1.3. This approach aims to align the volume of training data with the model’s parameter requirements. However, for some more complex IR tasks, an alternative solution might involve simplifying the task itself.

We focus on the task of developing scale-calibrated ranking models (Bai et al., 2023; Yan et al., 2022), where the ranking scores should not only generate an optimal order of documents, but also align with the labels of individual documents. For example, consider three documents A, B, and C labeled as 3, 2, and 1, representing “highly relevant,” “somewhat relevant,” and “not relevant,” respectively. A ranking model that scores these documents as  $-6$ ,  $-10$ , and  $-11$  successfully orders them but fails at scale calibration. Scale calibration is a nuanced IR task beyond ad-hoc retrieval because it requires the model to map the content and relation of query-document pairs to a specific scale, requiring the effective understanding of the scoring rubrics. Accurate scale-calibrated scores are easy to interpret, and enhance other IR tasks such as cut-off prediction and query performance prediction.

Scale calibration in neural ranking models (NRMs) faces pronounced data scarcity challenges. Obtaining datasets with labels on a meaningful scale (e.g., multi-level relevance judgement) presents a considerable challenge. For instance, large datasets

like MS MARCO (Bajaj et al., 2018) contain hundreds of thousands of queries but lack graded relevance, whereas datasets with detailed relevance judgments might only encompass hundreds of queries (Craswell et al., 2021). This limited availability of suitable training data severely restricts the development of effective scale-calibrated NRMs, which require more data than their non-calibrated counterparts due to the complexity of the task.

To address the data scarcity issue in developing scale-calibrated neural ranking models, we use zero-shot natural language explanations (NLE) from large language models (LLMs). This approach reduces the task’s complexity and, consequently, the volume of training data required. LLMs analyze the content and relationships within query-document pairs and generate NLEs. We also enhance the reliability of these explanations by employing Monte Carlo sampling to capture the uncertainty LLMs face when determining the relevance of these pairs. The synthesized NLEs, incorporating insights and uncertainties from LLMs, serve as inputs to enhance neural ranking models, enabling them to produce effective and calibrated ranking scores.

- *Contribution 4.1.* We extend the scale calibration task from feature-based learning-to-rank models, as discussed in previous works (Bai et al., 2023; Yan et al., 2022), to text-based neural ranking models, which offer improved usability, generalizability, and effectiveness but require more extensive data for training.
- *Contribution 4.2.* We introduce the use of zero-shot NLEs from LLMs to simplify the scale calibration task. This innovative approach complements existing methods for developing scale-calibrated ranking objectives (Yan et al., 2022).
- *Contribution 4.3.* We experiment with various types of explanations (literal and conditional) and algorithms for aggregating multiple Monte Carlo samples of Natural Language Explanations (NLEs). Our experiments show that this

approach results in a reduction of calibration error by 25% and 16% on TREC and NTCIR respectively. These datasets feature multiple levels of relevance labels, and our method achieves this improvement while maintaining or even enhancing the ranking performance of neural rankers.

## **1.5 Summary**

This dissertation investigates various techniques for utilizing explanations to improve the interpretability and effectiveness of information retrieval systems under data scarcity constraints. These constraints arise from limited resources for generating explanations as well as for conducting information retrieval. The proposals, techniques, and findings outlined in this dissertation provide a solid foundation for future research on integrating natural language explanations into information retrieval systems, aiming to improve both users' understanding and system performance.

## CHAPTER 2

### RELATED WORK

In this chapter, we provide a concise overview of related work to contextualize our studies in the subsequent chapters. We begin by reviewing existing neural methods for ad-hoc retrieval, which are the primary targets for our efforts to improve their interpretability and effectiveness — refer to Section 2.1. Following this, in Section 2.2, we introduce explainable information retrieval, focusing on prior efforts to enhance the interpretability of information retrieval systems. Lastly, we discuss recent approaches that incorporate natural language explanations and generative large language models, detailed in Section 2.3.

#### 2.1 Neural Ad-hoc Retrieval

Ad-hoc information retrieval involves finding resources relevant to a user query expressed in natural language. Prior to the emergence of deep neural networks, several seminal works established the foundation for information retrieval tasks. For instance, the Vector Space Model (VSM) (Salton et al., 1975) represents documents and queries as vectors in a high-dimensional space, enabling relevance calculations through geometric measures like cosine similarity. This model was enhanced by Latent Semantic Indexing (LSI) (Deerwester et al., 1990), which addressed some of VSM’s limitations by using dimensionality reduction in an effort to capture deeper semantic relationships between words. Subsequently, term matching models such as BM25 (Robertson et al., 1995) refined the assessment of term importance by incorporating term frequency and document length into their ranking functions, elements also utilized in

VSM. The query likelihood model (Ponte and Croft, 2017) further advanced retrieval accuracy with a probabilistic approach based on the likelihood of observing the query terms in each document. It is important to acknowledge that they represent only a subset of the influential work in the field, with many other important research also shaping our understanding of effective retrieval strategies. The field then transitioned to learning-to-rank approaches, utilizing machine learning algorithms ranging from linear models (Metzler and Bruce Croft, 2007) to tree-based methods (Burges, 2010), support vector machines (Joachims, 2002), and neural networks (Burges et al., 2005), which all depended heavily on delicate feature engineering. The introduction of Word2Vec (Mikolov et al., 2013) marked significant advances in learning abstract text representations, leading to subsequent neural text matching models (Dai et al., 2018; Guo et al., 2016; Pang et al., 2016; Xiong et al., 2017) that do not require feature engineering.

The advent of pre-trained language models (PLMs), exemplified by BERT (Devlin et al., 2019), has further revolutionized neural ad-hoc retrieval, making PLMs foundational to effective retrieval systems due to their superior language understanding and adaptability with minimal fine-tuning. Nogueira and Cho (2019) pioneered the fine-tuning of BERT with learning-to-rank objectives, demonstrating remarkable reranking performance. However, this “cross-encoder” method raised efficiency concerns due to the necessity of multiple forward passes during real-time reranking. To mitigate this, the “bi-encoder” approach was developed, allowing for the pre-computation and indexing of document representations offline; during online retrieval, only the query is encoded on-the-fly. This model has been incorporated into various ad-hoc retrieval methods including single-vector dense retrieval (Gao and Callan, 2021; Karpukhin et al., 2020; Xiong et al., 2020), multi-vector dense retrieval (Khattab and Zaharia, 2020), and learned sparse retrieval (Formal et al., 2021; Mallia et al., 2021; Yu et al., 2024).

As ad-hoc retrieval models improve, it is crucial to consider the underlying data scarcity challenges. To achieve the advance in retrieval performance, the data requirements have also increased substantially. The complexity of IR models has evolved from BM25 to learning-to-rank models and now to PLM-based models, expanding the data needs from none or tens to hundreds to even millions of queries. Despite the availability of large-scale training datasets like MS MARCO (Bajaj et al., 2018), it is essential to recognize that English-only ad-hoc retrieval focused solely on relevance does not cover all retrieval tasks. For instance, ad-hoc retrieval involving non-English languages, particularly low-resource languages, still faces significant data scarcity (Huang et al., 2023b). Moreover, the format of existing large-scale IR datasets often fails to support tasks with target beyond relevance. For instance, such datasets do not contain critical information on subtopic coverage and thus cannot be leveraged to enhance the diversification aspect of ad-hoc retrieval. Furthermore, it has been noted that PLM-based retrieval models trained on MS MARCO sometimes perform comparably to or even worse than BM25 in certain domains and tasks (Thakur et al., 2021).

We introduce three significant contributions to in this dissertation targeting to address data scarcity in neural ad-hoc retrieval. Chapter 4 describes a PLM-based framework specifically designed to enhance the diversification aspect of neural ad-hoc IR. Due to the limited amount of diversification data available, previous research has resorted to external query aspect generation systems, leaving the process ungrounded in the documents and unoptimized. Our method achieves state-of-the-art performance with just over a hundred training queries, thanks to an explanation-matching pre-training task on open-domain data. Chapter 5 discusses leveraging explanation-guided generative large language models to enhance synthetic query generation, a crucial method for overcoming data scarcity in training neural ranking models. Unlike previous works, our method focuses on generating hard negative training samples,

which have proven to be more effective and efficient during the training phase. Finally, Chapter 6 explores the use of automatically generated natural language explanations to augment scale calibration of neural ranking models. Previous work only addressed the calibration of smaller-scale learning-to-rank models due to data scarcity. We address this limitation by transforming the task into an easier one, effectively overcoming the challenges posed by scarce data.

## 2.2 Explainable Information Retrieval

Research in the field of explainability and interpretability of information retrieval systems predominantly concentrates on two distinct areas: explainable ad-hoc retrieval, and explainable recommendation and product search. The former focuses on clarifying how users’ queries match with text documents in terms of relevance. In contrast, recommendation and product search prioritize factors such as entity relationships and user purchase history over query-document matching for identifying user preferences and purchase behaviors (Ai et al., 2019a; Ai and Narayanan.R, 2021; Ai et al., 2019b; Carmel et al., 2020). As a result, the methods used to generate explanations in these areas differ significantly. This dissertation specifically addresses explainable ad-hoc retrieval. Within this domain, there are two main research objectives: enhancing *post-hoc interpretability*, which helps users better understand the outcomes of IR systems, and developing inherently interpretable IR models, which foster greater trust in these systems—termed *intrinsic interpretability*.

### 2.2.1 Post-hoc Interpretability

Post-hoc interpretability methods serve the purpose of explaining the decisions made by trained models. This approach proves advantageous when interpretability is required without necessitating modifications to or even understandings of existing models or services. Based on the the form of explanations provided, post-hoc inter-

pretability methods can be generally classified into feature attribution and natural language explanation.

- **Feature attribution.** Feature attribution methods, also known as feature importance or saliency methods, elucidate individual predictions by linking the model’s output to its input features, ranging from words and passages in text retrieval and ranking tasks to numeric and categorical features in learning-to-rank scenarios. In the model-agnostic realm, Local Interpretable Model-agnostic Explanations (LIME) is widely utilized (Ribeiro et al., 2016). LIME interprets black-box classifiers by training a surrogate model on perturbed samples to approximate local behavior, revealing feature significance through sparse linear models, and has been adapted to explain rankers by converting scores into probabilities (Singh and Anand, 2019; Verma and Ganguly, 2019). Conversely, model-introspective methods like Integrated Gradients (Sundararajan et al., 2017), DeepLIFT (Shrikumar et al., 2017), and DeepSHAP (Lundberg and Lee, 2017), use gradients or attention scores to assess feature importance, proving crucial in neural retrieval and Transformer-based rankers (Fernando et al., 2019; Purpura et al., 2021; Zhan et al., 2020). However, the efficacy of attention weights in providing genuine explanations is contentious (Bastings and Filippova, 2020; Bibal et al., 2022; Jain and Wallace, 2019; Wiegrefe and Pinter, 2019).
- **Natural Language Explanations.** Feature attribution methods often yield explanations that are difficult for non-experts to understand, limiting their utility to specialized audiences such as model developers and IR practitioners. Natural language explanations enhance user interaction by focusing on the usability of explanations and prioritizing user experience over the intricate details of model mechanics, making them broadly accessible and model-agnostic (Thomas et al., 2019). One strategy for providing these explanations involves clarifying query intent, as search systems may misinterpret user queries, especially with the application of

query expansion techniques. Describing how a system perceives a query can help users understand the results and refine their searches (Singh and Anand, 2020; Zhang et al., 2020). Another approach is explaining document relevance directly through natural language, which allows the use of information beyond the input features. Query-biased snippets, or document summaries, serve this purpose in search engines, providing insights into document relevance (Chen et al., 2020b; Tombros and Sanderson, 1998; Turpin et al., 2007). Rahimi et al. (2021) advocate for more concise, descriptive explanations that enable users to quickly and accurately discern document relevance.

### 2.2.2 Intrinsic Interpretability

In contrast to the pursuit of post-hoc interpretability, a different research direction focuses on intrinsic interpretability. This approach involves designing models that are inherently explainable, also known as interpretable-by-design (IBD) (Anand et al., 2022). The objective is to build models that possess interpretability from their inception. However, achieving complete transparency while maintaining competitive performance poses a significant challenge, especially for complex, nonlinear, and over-parameterized neural models. It is important to note that most existing IBD methods in the literature provide only partial interpretability.

- **Explainable-by-architecture.** Models that are inherently explainable generally employ two approaches: replacing black-box components with interpretable models (like decision trees) or operations (such as summation), or reducing feature complexity and making them more interpretable. For learning-to-rank (LTR) tasks with a limited number of numerical input features, creating transparent models is feasible due to the small, well-defined feature space. Common interpretable models used in this context include Generalized Additive Models (GAM) (Hastie and Tibshirani, 1986) and Decision Trees. GAM simplifies outputs as summations of

features, and Decision Trees use yes/no decisions (Lucchese et al., 2022; Zhuang et al., 2021). Neural ranking models, which incorporate numerous query-document term interactions like cosine similarities (Khattab and Zaharia, 2020; Pang et al., 2016), are often less interpretable. Query-document interaction functions can be used to improve interpretability, like DRMM (Guo et al., 2016) which uses matching histograms, or Transformer-Kernel (Hofstätter et al., 2020) which applies kernel-pooling techniques.

- **Rationale-based Methods.** These methods achieve inherent interpretability by generating intermediate rationales—extractive segments from input that significantly contribute to decision-making. Rationale-based methods typically involve a two-stage process: rationale extraction and prediction based on the extracted rationale. The rationale serves as a transparent explanation for the model’s decisions (DeYoung et al., 2020; Lehman et al., 2019; Zhang et al., 2021). In information retrieval tasks, advanced implementations like the Intra-Document Cascading Model address the ranking of long documents by selecting relevant passages (Hofstätter et al., 2021). Alternatively, models like Select-and-Rank (Leonhardt et al., 2023) and techniques leveraging the Gumbel-Softmax trick (Bang et al., 2021) or the information bottleneck concept (Jiang et al., 2021) allow for end-to-end training with discrete variable sampling, enhancing the interpretability of rationale extraction and overall model explainability.

We introduce two significant contributions in this dissertation to explainable information retrieval. Chapter 3 presents a post-hoc interpretability method that utilizes natural language explanations to enhance the understanding and organization of search engine result pages (SERPs). Unlike other post-hoc approaches, our work specifically addresses how explanations can effectively incorporate contextual information from other documents, reflecting the comparative nature of relevance ranking in ad-hoc retrieval. Chapter 4 details an intrinsic interpretability approach that em-

ploys extracted subtopics as intermediate rationales within an end-to-end framework, thus enhancing the effectiveness of diversified ranking tasks. Our method is distinctive compared to other rationale-based methods in that it leverages open-domain knowledge to provide weak supervision for the sub-task of rationale learning, offering a novel approach to this area of study.

### 2.3 Explanations and Generative Large Language Models

Recent advances in scaling pre-training and reinforcement learning from human feedback have significantly enhanced the development of generative large language models (LLMs). One key feature of these models is their ability to adapt to new language tasks with minimal input, relying solely on prompts and, optionally, a few training examples—all within the same context and without requiring parameter updates (Brown et al., 2020). However, despite their impressive capabilities, LLMs still face challenges with interpretability and tendencies to generate nonsensical or hallucinated content, raising intriguing questions about their inner workings and limitations.

To gain insights into how LLMs process language tasks, a promising approach is to enable LLMs to “explain themselves” (Lampinen et al., 2022; Nye et al., 2021; Wei et al., 2022b). Here, explanations broadly refer to the intermediate thought processes from input to output. This method involves augmenting each human-provided example (input and label) with natural language explanations, prompting the LLM to generate an explanation for its prediction. The advantages of this approach include simplifying complex tasks, mimicking human reasoning, and enhancing the credibility of model outputs through interpretable reasoning. At a higher level, these strategies represent two approaches to using explanations with LLMs: one focuses on improving model effectiveness through explanations (denoted as **Explanations to LLMs**), and the other on deriving explanations themselves (referred to as **Explanations by LLMs**). These methodologies are explored further in subsequent sections

### 2.3.1 Explanations to LLMs

Explaining tasks using explicit natural language instructions has proven effective for adapting large language models to new tasks (Liu et al., 2023). Additionally, decomposing the reasoning process into steps *for training examples* enhances the few-shot performance of LLMs. For instance, Nye et al. (2021) demonstrate this by using LLMs to predict the final outputs of Python programs through the prediction of intermediate computational results line-by-line. Similarly, the Chain-of-Thought (CoT) prompting method, proposed by Wei et al. (2022b), involves generating a series of sentences that map the reasoning process from input to final output as explanations, before arriving at the final prediction. This method has shown consistent superiority over standard prompting in arithmetic, commonsense, and symbolic reasoning tasks. The effectiveness of CoT prompting is further validated in multimodal scientific question answering contexts (Lu et al., 2022). Additionally, Wang et al. (2022b) introduce a voting mechanism to select the most consistent answer from different reasoning paths, and Li et al. (2022b) employ a verifier to assess the quality of each path and guide the voting. Lampinen et al. (2022) provide a comprehensive study on the impact of post-hoc explanations (provided after the prediction, unlike CoT) in in-context learning, revealing that explanations significantly enhance performance, especially when they are fine-tuned, indicating that some explanations are more beneficial than others. Even un-tuned explanations still yield a positive effect, though primarily in larger LMs.

### 2.3.2 Explanations by LLMs

Instead of using explanations to achieve better LLM generations, another line of research focuses on examining the actual explanations generated by LLMs, their characteristics, and their utility. Ye and Durrett (2022a) evaluate explanations for in-context learning on two textual reasoning tasks—question answering and natu-

ral language inference. They find that while including explanations in the prompts leads to moderate accuracy improvements, the explanations generated by LLMs often do not align with the models’ predictions (being inconsistent or unfaithful) and are not always factually grounded in the input. Similarly, Turpin et al. (2023) show that Chain-of-Thought (CoT) explanations are systematically unfaithful, as LLM behavior can be predictably influenced by biased features in their inputs that are not mentioned in the explanations. Despite criticisms, LLM-generated explanations have been utilized as silver training data. In some instances, these explanations are fed back into LLMs to enhance their reasoning capabilities in a bootstrapping manner (Ma et al., 2023a; Zelikman et al., 2022) or used to guide the training of smaller models (Li et al., 2022a). In information retrieval, Ferraretto et al. (2023) propose using GPT-3 generated explanations of relevance and irrelevance to train a T5-based passage ranker. The authors demonstrate that incorporating these explanations significantly enhances the effectiveness of the ranker, particularly in the regime of data scarcity.

We introduce two novel and significant contributions in this dissertation to the enhancement of neural ranking models using natural language explanations with large language models. Chapter 5 explores explanation-guided prompting—an approach where LLMs generate synthetic queries to produce highly informative training data. This method ensures that the contrasting documents serve as hard negatives, addressing a gap in previous works that leverage LLMs for synthetic query generation. Chapter 6 discusses adopting an explanation-by-LLMs approach, which utilizes the reasoning and uncertainty information inherent in natural language explanations to calibrate neural ranking models more effectively. Unlike previous research that focused solely on the calibration of learning-to-rank models, our method offers a more effective, generalized and applicable solution to this critical task.

## CHAPTER 3

# LISTWISE MODEL-AGNOSTIC EXPLANATION GENERATION FOR SERPS

To enhance user comprehension of why documents are retrieved, search engine result pages (SERPs) typically display details such as the document title, URL, and a snippet—a brief, 2- or 3-line, query-focused summary. Nevertheless, Thomas et al. (2019) found that fewer than 1% of users grasped the topical diversity of the search results from this format, indicating substantial potential for improvement in how search results are explained.

In pursuit of this improvement, Rahimi et al. (2021) introduced a text generation model that provides aspect-oriented explanations of documents within search results. Their findings indicate that these explanations significantly aid users in navigating a ranked list of search results, enhancing inter-annotator agreement on document relevance by 37% and reducing the time taken to identify relevant documents by 22%. Similarly, other studies have confirmed the benefits of aspect-oriented explanations in helping users efficiently find pertinent information (Haag et al., 2014; Iwata et al., 2012b).

However, a limitation of these approaches is that they explain each document in isolation, without considering its context relative to other documents in the list. This often results in explanations that are identical or highly similar, as all respond to the same query. In Figure 3.1, documents  $d_2$  and  $d_3$  are explained with the same query aspect “history”: correct but not helpful. Another drawback is that explanations in isolation tend to be generic and lack depth, ultimately failing to effectively aid users in differentiating between documents.

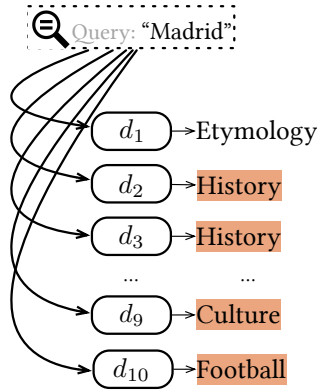
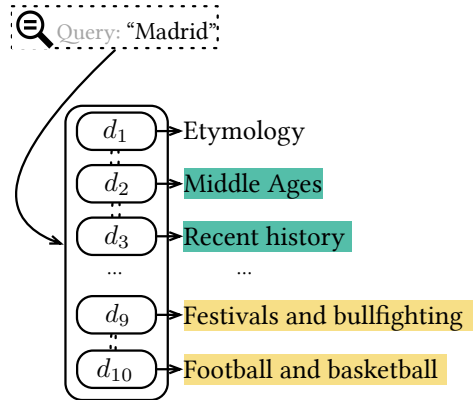
**Independent and Single-aspect Explanations:****Joint and Multi-aspect Explanations :**

Figure 3.1: Explaining a search engine result page (SERP) in different settings.

We refine the task of search result explanation (SeRE) by generating concise relevance explanations for each document in a search results list. These explanations are designed to: (1) describe the query aspects at the phrase level; (2) cover multiple aspects; (3) derive solely from the documents’ content; and (4) exhibit diversity. Figure 3.1 illustrates the distinction between traditional single-aspect explanations and our proposed multi-aspect, context-aware approach. For this purpose, we have developed weakly-labeled datasets from the English Wikipedia to facilitate training neural network-based language generation models. Additionally, we adapt the MIMICS dataset (Zamani et al., 2020), originally compiled from real query logs of the Microsoft Bing search engine, to evaluate the generated explanations of web search results in terms of both the relevance and the topical diversity.

To address the SeRE task, we introduce the LiEGe (**L**istwise **E**xplanation **G**enerator) model, which *jointly* produces aspect-oriented explanations for *all* documents in a search results list. This model employs a novel Transformer-based encoder-decoder architecture to provide multi-granular semantic representations of documents and their tokens within the entire context of search results, utilizing various interaction signals. Experimental results confirm the LiEGe model’s ability to generate more

precise and diverse aspect-level explanations, underscoring its effectiveness for the SeRE task.

*The work described in this chapter, namely “Towards Explainable Search Results: A Listwise Explanation Generator”, was published in SIGIR 2022 (Yu et al., 2022). I was the lead author responsible for creating datasets, designing model architectures and conducting experiments.*

### 3.1 Multi-aspect and Listwise Search Result Explanation

It is common that web documents cover multiple query aspects (Carterette and Chandar, 2009), something that is often overlooked in existing work on query aspect generation (Hashemi et al., 2021; Rahimi et al., 2021). To address this unexplored challenge in the SeRE task, we introduce two generation strategies for *multi-aspect* documents under the joint explanation setting: (1) **comprehensive explanation generation** (CEG), where all query aspects covered by each document in the ranked list are considered as explanations; and (2) **novelty explanation generation** (NEG), where the explanation for each document describes the novel relevant information of the document with respect to the documents preceding it in the list. One type of explanation may be more suitable than the other for a particular search task or search device.

Either type of explanation requires its own training and test set. In practice, these two tasks take the same set of inputs (a query and a list of documents), but have different outputs. To the best of our knowledge, there is no public dataset for listwise content-based explanation of search results. We thus adapt existing datasets for other similar tasks to the SeRE task. In the following, we describe how those datasets are built and processed.

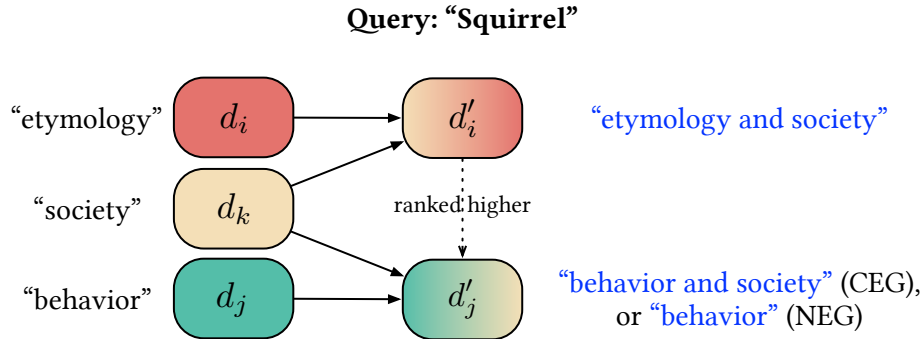


Figure 3.2: An example of creating multi-aspect documents from the English Wikipedia for the CEG and NEG task.

### 3.1.1 Wikipedia as a Weakly Labeled Dataset

One approach to constructing a large-scale training dataset automatically is to treat each Wikipedia article as a search result. Here, the article title, typically an entity name, serves as the query, while the content of each section acts as a document retrieved in response to the query. The section headings provide aspect-based explanations of how the content relates to the query and other sections. Notably, most sections in Wikipedia (referred to as documents in this analogy) focus on a single aspect of the query, reflecting the topical organization into sections by human experts over several iterations. We designate this dataset as **Wiki-SA**, where “SA” stands for single-aspect, akin to the Wiki dataset used by Rahimi et al. (2021) and the WikiOG dataset (Zhang et al., 2019b). However, Wiki-SA is not ideal for training or evaluating explanation generation for documents that address multiple query aspects. Similarly, ClueWeb and MS MARCO, which are used for pointwise explanation, are unsuitable as they contain few or no documents annotated with multiple query aspects.

To generate a substantial amount of training data from Wikipedia for the CEG and NEG settings, we introduce a method called **fusing**, which intentionally creates overlapping content between documents in a search result. This process enables some documents to cover multiple aspects of the query, as illustrated in Figure 3.2. Specif-

Table 3.1: Statistics of created datasets for the new SeRE task.

Dataset	Wiki			MIMICS	
	-SA	-CEG	-NEG	-CEG	-NEG
# Queries	39,287	39,287	39,287	1,992	1,992
# Documents per query	7.5	5.4	5.4	5.2	3.0
# Words per document	184.9	344.2	344.2	54.0	54.0
# Words per Explanation	2.7	5.1	3.9	1.9	1.4

ically, we randomly select three documents,  $d_i$ ,  $d_k$ , and  $d_j$ , from the same Wikipedia article. These documents initially have aspect-based explanations  $e_i$ ,  $e_k$ , and  $e_j$ , respectively. To create overlapping content, we combine  $d_i$  and  $d_k$  to form a new document  $d'_i$ , and  $d_j$  and  $d_k$  to form  $d'_j$ . The order of concatenation is randomized to ensure that the explanation model can recognize novel content without relying on the position of information within the documents. In Wiki-CEG, the explanations for  $d'_i$  and  $d'_j$  are labeled as “ $e_i$  &  $e_k$ ” and “ $e_j$  &  $e_k$ ”, respectively. For Wiki-NEG, where the ranking of documents affects the ground-truth explanations, we assume  $d'_i$  is ranked higher than  $d'_j$ . Thus,  $d'_i$  and  $d'_j$  are explained as “ $e_i$  &  $e_k$ ” and “ $e_j$ ” respectively, with the rationale that  $e_i$ , though present in  $d'_j$ , is already covered by  $d'_i$ .

From a Wikipedia article with  $s$  sections, our fusing method produces approximately  $\lfloor s/3 \rfloor \times 2$  documents. Any remaining sections—up to two—are retained in the dataset as single-aspect documents.

All three datasets—Wiki-SA, Wiki-CEG, and Wiki-NEG—are constructed from the same pool of Wikipedia articles, chosen for having at least six sections, each ranging from 128 to 256 words. This word count ensures each section is substantial enough to stand as a document. The upper limit of 256 words accommodates the 512-token input limit of BERT, particularly relevant when sections are concatenated in the Wiki-CEG and Wiki-NEG datasets. The requirement of at least six sections guarantees that the fused datasets include at least four documents in each search result list. After applying these criteria, 39,287 Wikipedia articles qualified for inclusion.

Each article represents an instance of a query and search result list  $(q, R)$  across the Wiki-SA, Wiki-CEG, and Wiki-NEG datasets. These instances are divided into training, development, and test sets in an 80%/10%/10% ratio, respectively. The development set is utilized for tuning hyperparameters and determining the point of early training stop.

The Wiki-CEG and Wiki-NEG datasets differ from Wiki-SA in both the number of documents per search result and the length of those documents due to the fusing process. Table 3.1 details these statistics, including the number of instances, the average number of documents in search results, the average document length, and the average length of explanations.

### 3.1.2 Adapting MIMICS Datasets for Evaluation

MIMICS (Zamani et al., 2020) is a collection of datasets for search clarification built from real search queries sampled from the Bing query logs. Besides its real queries and search results, MIMICS has another advantage over the Wiki datasets: two documents in a search result list can cover the same query aspect without having exactly the same content for the common aspect. This property makes the evaluation of CEG and NEG more realistic. Each clarification in MIMICS consists of a query, a clarifying question, up to five candidate answers for the clarifying question which are aspects of the query, and the top-10 documents retrieved by Bing. For SeRE, we need to have gold explanations for documents based on query aspects. However, MIMICS does not contain aspect-level relevance information. In other words, the top-10 documents retrieved with respect to a query as well as the query aspects are provided, but which documents are relevant to which query aspects are not specified. To adapt MIMICS for the SeRE task, we make the conservative assumption that a document is considered relevant to a query aspect only if it contains the aspect terms.

We thus obtain high quality labels for aspect-level relevance, at the cost of missing some relevance labels.

We chose the ClickExplore version of MIMICS as it contains the largest number of unique queries. We perform the following processing steps. (1) Query terms are removed from aspects, as repeating query terms in explanations does not provide additional information; (2) The concatenation of a document’s heading and snippet is used as the document content (Hashemi et al., 2021). Note that full document contents are not released in the MIMICS dataset; (3) Documents that are not labeled as relevant to any query aspects are removed; (4) Query aspects that are not associated with any documents in a search result list are also removed. If the number of remaining query aspects is less than three, the query is removed; (5) Queries whose clarification has engagement level below 4 (out of 10) are removed. Engagement level indicates the quality of clarification and query aspects perceived by users (Zamani et al., 2020). In the end, we acquired 1,992 queries from the original dataset, which are split into train/test set evenly. Similar to the Wiki datasets, we create two variants from MIMICS to separately evaluate CEG and NEG. For **MIMICS-CEG**, the explanation of a document contains all associated query aspects. For **MIMICS-NEG**, a document’s explanation only contains aspects that are novel considering its preceding documents. In cases that a document contains no novel aspects, we discard it. Statistics of these datasets are also reported in Table 3.1.

### 3.2 Listwise Explanation Generator (LiEGe)

Given a query  $q$  and the top-ranked documents  $R = \{d_1, \dots, d_k\}$  retrieved or reranked, the objective of LiEGe is to generate an explanation for each document in  $R$ . These explanations articulate the specific information each document provides in relation to the query  $q$ . Each explanation highlights the query aspect(s) covered by a document  $d_i$ , either in a **novelty-** or **comprehensive-**based manner.

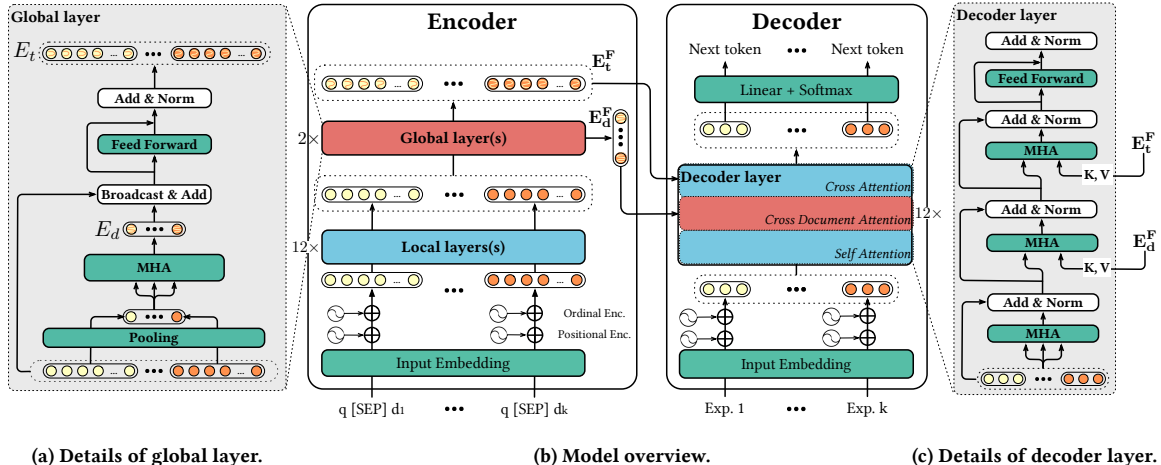


Figure 3.3: The architecture of LiEGe. Panel (b) presents a high-level overview of the encoder-decoder structure, with specific modifications for listwise explanation generation highlighted in red. Detailed descriptions of these changes are provided in panels (a) and (c), respectively.

In order to describe the relevant and distinct information that each document in a search result list provides, a document needs to be encoded with respect to the query and with respect to the other documents in the list. The former is required to capture the relevant part of a retrieved document, as a small portion of a document may be all that is related to the query. The latter part of encoding exploits cross-document interactions so that generated explanations can help users distinguish the differences between documents in the search result. The encoded input is then passed to the decoder to generate explanations. Figure 3.3 (b) shows the architecture of LiEGe.

### 3.2.1 Input Representation

Given the query  $q$  and the retrieved documents  $R = \{d_1, \dots, d_k\}$ , each document  $d_i$  is concatenated with the query, separated by a specific token. The resulting sequence  $q - d_i$  is then tokenized and adjusted to a uniform length of  $l$  tokens through truncation or padding. Positional and segment embeddings are integrated into the input token embeddings to enrich the representation. To generate novelty explanations, the model incorporates the ranks of the documents in the search result list, enabling

it to identify the novelty of a document relative to its predecessors. To achieve this, ordinal encodings with the same dimension  $m$  are added to the input embeddings. These encodings are generated by an embedding function  $P$ , similar to that used by Pang et al. (2020), which encodes the absolute rank of each document  $d_i$  into an embedding vector  $p_i \in \mathbb{R}^m$ . This rank embedding  $p_i$  is subsequently added to every token embedding of  $d_i$ . Consequently, the query-document pair  $(q, d_i)$  is represented as  $X_i \in \mathbb{R}^{l \times m}$ , and the entire list  $R$  is represented by the following equation:

$$X = [X_1, \dots, X_k] \in \mathbb{R}^{k \times l \times m}, \quad (3.1)$$

### 3.2.2 Encoder

The encoder consists of a stack of local and global attention layers. A local attention layer updates token representations based on intra-document self-attention, while a global attention layer updates representations based on inter-document attention. In other words, local layers perform per document computation while global layers perform per result list computation.

We denote the input *token* representations to the  $h$ -th transformer layer of the encoder as  $E_t^{(h)} \in \mathbb{R}^{k \times l \times m}$ . Note that  $E_t^{(1)} = X$  as defined in Eq. 3.1. The output token representations of the layer are denoted by  $E_t^{(h+1)} \in \mathbb{R}^{k \times l \times m}$ , which constitute the input to the next layer, if any. A global layer has one additional output compared to a local one. The additional output is *document* representations for all documents in the result list, denoted by  $E_d^{(h+1)} \in \mathbb{R}^{k \times m}$ . These document representations will also be used in the decoder.

**Local layers** perform multi-head self attention on the token sequence from a query-document pair (intra-document), similar to the encoder layers in Transformer. The contextualized representations are then linearly transformed. Residual connection and layer normalization are applied for each of the layers (Vaswani et al., 2017).

The function of a local layer can be formalized as:

$$L_t = \text{LN}(E_t^{(h)} + \text{MHA}(E_t^{(h)}, E_t^{(h)}, E_t^{(h)})), \quad (3.2)$$

$$E_t^{(h+1)} = \text{LN}(L_t + \text{FFN}(L_t)), \quad (3.3)$$

where  $\text{LN}(\cdot)$  is layer normalization, and  $\text{FFN}(\cdot)$  stands for position-wise feed-forward networks.

**Global layer** first generates dense document embeddings by pooling, where each document in the search result is represented with a single embedding of dimension  $m$ . We consider two pooling strategies: (1) applying multi-head pooling (Liu and Lapata, 2019) to learn a weighted average of the embeddings of the document’s tokens; and (2) simply taking the embedding of the first token ([CLS]) as the representation of the entire sequence. After acquiring a list of dense embeddings for documents via pooling, multi-head self-attention is applied across documents in the search result list. The output is still a list of dense embeddings, where each document embedding is contextualized based on the other documents in the search result. In order to propagate information from document interactions at the document granularity to the token granularity, the contextualized embedding of a document is added to the embedding of each of its tokens. We refer to this function as *broadcast & add*. Figure 3.3 (a) shows the architecture of a global layer.

Specifically, the outputs of a global layer are calculated as:

$$E_d^{(h)} = \text{Pool}(E_t^{(h)}), \quad (3.4)$$

$$E_d^{(h+1)} = \text{MHA}(E_d^{(h)}, E_d^{(h)}, E_d^{(h)}), \quad (3.5)$$

$$\hat{E}_t[i, j] = E_t^{(h)}[i, j] + E_d^{(h+1)}[i], \quad (3.6)$$

$$E_t^{(h+1)} = \text{LN}(\hat{E}_t + \text{FFN}(\hat{E}_t)), \quad (3.7)$$

where  $E_t^{(h)}[i, j]$  is the embedding of the  $j$ -th token of document  $i$  in the input token representations  $E_t^{(h)}$ .  $E_d^{(h+1)}[i] \in \mathbb{R}^m$  is the representation of the  $i$ -th document contextualized based on all documents in the search result. Eq. 3.6 shows the broadcast & add operation, which is performed for all tokens of all documents. Therefore, information from all documents is considered and appropriately reflected in the representation of every document token.

**Encoder outputs.** We denote the output token representations from the *final* encoder layer as  $E_t^F$ , and the contextualized document representations from the *final global* layer in the encoder as  $E_d^F$ . Note that  $E_t^F$  can be the output of a global or a local transformer layer, depending on the model composition.  $E_t^F$  and  $E_d^F$  are used as inputs to the decoder.

### 3.2.3 Decoder

Each layer in the decoder of the Transformer contains two attention sub-layers: a self-attention sub-layer and a cross-attention (also called “encoder-decoder attention”) sub-layer. The purpose of self-attention in the decoder is to effectively use the already generated text for the prediction of the next token. The decoder uses cross-attention to utilize the encoder representations of input for identifying which part of the input sequence it should focus on to predict the next token. To avoid confusion, we refer to this sub-layer as *cross-token* attention.

**Cross-document attention sub-layer.** We propose a cross-document attention sub-layer, which is placed between self-attention and cross-token sub-layers in each decoder layer. The details of a decoder layer in LiEGe are depicted in Figure 3.3 (c). With cross-document attention, the representation of each token generated so far is updated by information from the contextualized embeddings of all documents in the search result from the encoder ( $E_d^F$ ). Attention to global information  $E_d^F$  helps the model to identify which specific query aspect should be generated. Then, through

attention to local document information ( $E_t^F$ ), the model identifies aspect-related part(s) of the document to be used for generation of the next token.

Input to the  $h$ -th layer of the decoder is denoted as  $D_t^{(h)} \in \mathbb{R}^{k \times l' \times m}$ , where  $l'$  is the maximum output length (during training) or the length of the currently generated sequence (during inference). The function of a decoder layer with a cross-document attention sub-layer is formally formulated as follows.

$$D_t = \text{LN}(D_t^{(h)} + \text{MHA}(D_t^{(h)}, D_t^{(h)}, D_t^{(h)})), \quad (3.8)$$

$$D_t = \text{LN}(D_t + \text{MHA}(D_t, E_d^F, E_d^F)), \quad (3.9)$$

$$D_t = \text{LN}(D_t + \text{MHA}(D_t, E_t^F, E_t^F)), \quad (3.10)$$

$$D_t^{(h+1)} = \text{LN}(D_t + \text{FFN}(D_t)). \quad (3.11)$$

After the final decoder layer, a linear and a softmax layer predict the next token to be generated for the explanation of each document. At inference time, generation repeats until either the end-of-sentence token is generated or the maximum output length is reached.

### 3.2.4 Training of LiEGe

Instead of using random query-document pairs for mini-batch training, we group documents from the same search result to leverage their interactions. We define *group size*  $k$  as the maximum number of documents considered in a search result for a query. During training, search results containing fewer than  $k$  documents are padded to maintain a constant list size of  $k$ . We also define *batch size*  $b$  as the number of groups (SERPs) included in a batch. Thus, each batch inputs  $b \times k$  documents into the model, with  $b$  groups processed in parallel. We implement a global document mask to ensure that (1) padded documents are excluded from local and global attention

calculations; and (2) global attention is confined to documents within the same group. In our experiments, we accommodate up to 10 documents per query ( $k = 10$ ), which aligns with the number typically displayed on the first page of results by modern search engines.

In line with prior research on sequence-to-sequence transduction, we employ the cross-entropy of the predicted and gold probability distributions at each position as our loss function to guide parameter optimization (Lewis et al., 2020). This loss function is computed across all positions in the sequences that are generated.

### 3.3 Experimental Setup

In this section, we detail the baseline approaches and metrics employed to evaluate the quality of explanations generated by LiEGe. We previously described the evaluation datasets, including Wiki-CEG, Wiki-NEG, MIMICS-CEG, and MIMICS-NEG, in Sections 3.1.1 and 3.1.2.

#### 3.3.1 Competing Methods

To the best of our knowledge, there are no existing models specifically designed for listwise content-based explanation of search results. Therefore, for the evaluation of LiEGe, we adapt several representative models from related tasks to the SeRE context and compare their performance. The models included in our comparison are:

- Unsupervised models: **TextRank** (Mihalcea and Tarau, 2004), **TS-TextRank** (Haveliwala, 2002) (a variation of TextRank that incorporates topic-sensitive PageRank), **KeyBERT** (Grootendorst, 2020), and **KeyBERT-MMR** (an adaptation of KeyBERT that generates novelty explanations using maximal marginal relevance (Carbonell and Goldstein, 1998)).
- Supervised models: **GenEx** (Rahimi et al., 2021), **NMIR** (Hashemi et al., 2021), **HiStGen** (Zhang et al., 2019b), **BART** (Lewis et al., 2020), and **LiEGe**.

Note that LiEGe, our best-performing model, uses weights initialized from BART and is configured by default with 12 local layers followed by 2 global layers. It utilizes multi-head pooling (8 attention heads) for the dense embeddings of documents and incorporates ordinal encoding. To provide a fair comparison, BERT and LiEGe (BERT) serve as additional baselines alongside the GenEx and KeyBERT models, which are also based on a pre-trained BERT. This setup ensures that any performance gains observed with LiEGe are not merely due to the pre-trained decoder from BART.

### 3.3.2 Evaluation Metrics

We primarily use the BLEU  $F_1$  metric (Papineni et al., 2002), reporting BLEU-1 as B-1 and the weighted geometric mean of BLEU- $k$  ( $k=1,2,3,4$ ) as BLEU. Additionally, we report ROUGE-1  $F_1$  and ROUGE-L  $F_1$  (Lin, 2004a) as R-1 and R-L, respectively. For evaluating multi-aspect explanations, we treat different aspects as multiple references in the computation of the BLEU and ROUGE metrics. To assess the *semantic* similarity between generated and ground-truth explanations, we employ BERTScore (Zhang et al., 2019c). We report micro-averaged BLEU, ROUGE, and BERTScore across all document-explanation pairs in a test set. Beyond accuracy-based metrics, we also measure the diversity of generated explanations within a search result list. This diversity metric, denoted as Div, is calculated as the average semantic similarity (using BERTScore) of all explanation pairs generated for a search result list. A lower Div score indicates greater diversity among the explanations provided for a search result list.

We use t-test with Bonferroni correction for statistical significance test at the level of 95%. Statistical significant improvements of LiEGe over *all* baselines are marked with \* in the result tables.

Table 3.2: Results for comprehensive explanation generation on the Wiki-SA dataset (single-aspect).

<b>Metric</b>	BLEU	B-1	R-1	R-L	BERTScore	Div ( $\downarrow$ )
TextRank	0.12	12.58	15.74	13.84	37.97	62.02
TS-TextRank	0.23	11.19	13.91	12.21	37.65	62.75
BERT-LIME	0.30	6.50	8.31	7.93	42.06	60.07
KeyBERT	2.11	13.81	16.87	16.40	46.04	58.43
GenEx	6.55	25.56	21.06	20.94	54.85	53.20
HiStGen	8.86	22.59	17.49	17.23	53.92	48.29
BERT	17.28	40.14	39.71	39.45	65.60	45.29
LiEGe (BERT)	19.27	41.40	40.76	40.55	65.98	45.03
BART	18.51	42.13	42.16	41.90	67.03	45.23
LiEGe	<b>21.97*</b>	<b>45.56*</b>	<b>45.84*</b>	<b>45.65*</b>	<b>69.01*</b>	<b>44.47*</b>

Table 3.3: Results for comprehensive explanation generation on the Wiki-CEG dataset (multi-aspect).

<b>Metric</b>	BLEU	B-1	R-1	R-L	BERTScore	Div ( $\downarrow$ )
TextRank	0.11	16.59	14.91	12.73	42.35	70.33
TS-TextRank	0.26	14.01	12.47	10.74	41.83	71.08
BERT-LIME	0.16	7.10	8.63	8.18	43.58	68.56
KeyBERT	1.52	15.11	12.98	12.51	49.90	66.98
GenEx	0.30	11.72	6.41	6.37	47.16	69.72
HiStGen	11.73	42.47	39.30	34.90	64.10	55.82
BERT	15.61	48.86	46.54	41.28	67.77	53.46
LiEGe (BERT)	17.36	50.46	47.69	42.52	68.40	53.00
BART	16.53	49.54	47.42	42.23	68.33	53.42
LiEGe	<b>18.79*</b>	<b>51.72*</b>	<b>49.37*</b>	<b>44.32*</b>	<b>69.46*</b>	<b>52.73*</b>

### 3.4 Experimental Results and Analysis

In this section, we discuss the results, focusing on comparisons between (1) LiEGe and other pointwise explanation methods, emphasizing the advantages of listwise explanations, which consider context; and (2) the impact of pretraining with Wiki on the quality of explanations in real web search settings. Additionally, we conduct an ablation study to analyze the significance of specific design elements within LiEGe.

### 3.4.1 Comprehensive Explanation Generation on Wiki

Performance of LiEGe and baseline models on the Wiki-SA and Wiki-CEG datasets is detailed in Table 3.2 and 3.3, respectively. The first block of these tables lists the unsupervised models—TextRank, TS-TextRank, BERT-LIME, and the BERT-based models KeyBERT—as well as GenEx, which, although trained on a dataset similar to Wiki-SA, was not further fine-tuned on Wiki-SA or Wiki-CEG. Consequently, GenEx performs best on Wiki-SA within this category but shows a significant drop in performance on Wiki-CEG due to its design to generate a single concise explanation per document, rather than a list of aspects. Other baseline models, which rank terms or phrases in a document, can adapt to our SeRE setting by selecting multiple outputs for each document. TextRank and TS-TextRank focus on extracting keywords, whereas KeyBERT identifies coherent noun phrases. The BLEU performance of TextRank and TS-TextRank is notably lower than that of KeyBERT because the BLEU metric also considers higher-order  $n$ -grams (B-2, B-3, and B-4).

The second block features HiStGen alone, differentiated from the first block by being trained on our datasets prior to testing. We modified HiStGen by removing the review mechanism (Chen et al., 2018) on Wiki-CEG to allow duplicate terms across different documents’ outputs, aligning with the ground-truth labels which share common query aspects and thus terms.

The third block includes BERT and LiEGe (BERT), while the final block presents BART and the complete version of LiEGe. In all evaluations, LiEGe significantly outperforms its base models. LiEGe’s improvements over pointwise explanation models like GenEx, BERT, and BART emphasize the effectiveness of listwise encoding and explanation in the SeRE context.

Table 3.4: Results for novelty explanation generation evaluated on Wiki-NEG.

	BLEU	B-1	R-1	R-L	BERTScore	Div ( $\downarrow$ )
KeyBERT	1.65	12.03	11.89	11.51	46.04	60.47
KeyBERT-MMR	1.77	12.90	12.83	12.30	47.14	58.62
HiStGen	8.31	29.47	25.64	23.98	56.65	51.68
BERT	10.75	34.50	31.69	29.71	59.47	49.66
LiEGe (BERT)	13.35	36.86	33.91	31.62	60.13	47.46
BART	11.84	35.94	33.80	31.84	60.82	49.38
LiEGe	<b>15.06*</b>	<b>39.45*</b>	<b>38.02*</b>	<b>35.80*</b>	<b>62.74*</b>	<b>47.13*</b>

### 3.4.2 Novelty Explanation Generation on Wiki

Results for the Wiki-NEG dataset are presented in Table 3.4. Initially, we note that enhancing KeyBERT with the MMR (Maximal Marginal Relevance) component improves its performance by promoting novelty; it penalizes phrases similar to those selected for earlier documents. HiStGen also effectively generates novelty explanations by incorporating a review mechanism (Chen et al., 2018). LiEGe emerges as the top-performing model in the novelty setting of SeRE. Although listwise modeling of search results has proven effective for comprehensive explanation generation as shown in Table 3.3, it is even more crucial for generating novelty explanations, where leveraging information from preceding documents is essential. Comparisons in Tables 3.4 and 3.3 reveal that LiEGe achieves higher percentages of improvement over baselines in the Wiki-NEG dataset than in the Wiki-SA and Wiki-CEG datasets.

### 3.4.3 Explanation Generation on MIMICS

Performance of LiEGe and baseline models on the MIMICS-CEG and MIMICS-NEG datasets is detailed in Tables 3.5 and 3.6. These tables illustrate the results of the LiEGe and BART models when pre-trained on the Wiki dataset and then fine-tuned on the corresponding MIMICS dataset (indicated with “Wiki”), as well as their performance when trained exclusively on the MIMICS dataset.

Table 3.5: Results for CEG evaluated on MIMICS-CEG.

	BLEU	B-1	R-1	R-L	BERTScore	Div ( $\downarrow$ )
TextRank	0.12	8.62	12.03	11.36	34.68	78.86
TS-TextRank	0.19	8.27	11.76	11.23	34.70	78.45
BERT-LIME	0.05	1.47	0.29	0.29	36.65	78.28
KeyBERT	0.79	9.36	12.81	12.61	39.46	77.00
GenEx	0.35	3.92	3.23	3.23	40.45	75.93
NMIR	0.03	3.34	6.91	6.28	32.19	99.70
HiStGen	19.82	39.65	31.14	30.56	55.60	64.16
BART	41.59	59.90	51.76	51.53	68.50	59.25
LiEGe	39.25	62.85	57.02	56.50	71.06	57.73
BART (Wiki)	45.61	63.65	59.06	58.10	72.13	57.52
LiEGe (Wiki)	<b>49.11*</b>	<b>68.91*</b>	<b>64.96*</b>	<b>64.09*</b>	<b>76.37*</b>	<b>56.34*</b>

Table 3.6: Results for NEG evaluated on MIMICS-NEG.

	BLEU	B-1	R-1	R-L	BERTScore	Div ( $\downarrow$ )
KeyBERT	0.79	7.00	10.43	10.42	37.29	75.34
KeyBERT-MMR	0.85	7.54	11.25	11.15	39.94	72.66
HiStGen	13.14	23.21	18.29	18.29	53.47	63.85
BART	25.49	44.74	42.08	41.99	64.93	55.28
LiEGe	33.17	55.94	52.83	52.67	71.26	51.39
BART (Wiki)	29.10	49.98	47.46	47.36	67.51	51.51
LiEGe (Wiki)	<b>37.00*</b>	<b>61.02*</b>	<b>59.02*</b>	<b>58.87*</b>	<b>75.40*</b>	<b>50.68*</b>

To the best of our knowledge, NMIR (Hashemi et al., 2021) is the state-of-the-art model for generation of query intents/aspects. Using NMIR for aspect-oriented explanation, however, generates explanations with the least amount of diversity compared to other baselines (Table 3.5). The main reason for this observation is that documents in the same cluster share the same explanation. This demonstrates that NMIR, and thus existing query intent generation models, do not address SeRE.

LiEGe consistently outperforms all baselines and its counterpart base model in both CEG and NEG settings, with the sole exception being its BLEU performance compared to BART in the comprehensive setting when both models are only trained on the MIMICS-CEG dataset (Table 3.5, row 9-10). This disparity may be attributed

to the relatively small size of the MIMICS data, which might be insufficient for effectively training the additional parameters in LiEGe compared to BART. This effect becomes more apparent when comparing the performance of BART (Wiki)/LiEGe (Wiki) against BART/LiEGe in both CEG and NEG settings. Specifically, pre-training on Wiki results in 9.7% and 25.1% improvements in BLEU for BART and LiEGe, respectively, in the comprehensive explanation generation (EG) setting, and 14.2% and 11.5% improvements in the novelty EG setting. These results confirm that the knowledge acquired from pre-training on the Wiki dataset can effectively translate to the real-world web data of the MIMICS datasets, substantially mitigating the data scarcity challenge in generating effective and concise natural language explanations in web search contexts.

### 3.4.4 Ablation Study

We train and test variants of LiEGe, leaving one component out at a time, on Wiki-CEG and Wiki-NEG separately. The ablated versions are as follows.

- **LiEGe w/o OE** does not add ordinal encodings to token embeddings in the encoder or the decoder.
- **LiEGe w/o MHP** uses the [CLS] token embeddings in each layer as pooled document representations, instead of using multi-headed pooling.
- **LiEGe w/o BA** does not add contextualized document embeddings to the embeddings of their tokens. More specifically, Eq. 3.10 is skipped and Eq. 3.11 becomes  $E_t^{(h+1)} = \text{LN}(E_t^{(h)})$ . A global layer thus only generates contextualized document embeddings as its output, and the final token embeddings from the encoder are not impacted by information from cross-document interactions.

Table 3.7: Performance of ablated variants of the LiEGe model. Symbol  $\nabla$  shows statistical significant differences comparing with LiEGe.

<b>Dataset</b>	Wiki-CEG		Wiki-NEG	
<b>Metric</b>	BLEU	R-L	BLEU	R-L
BART	16.53	42.23	11.84	31.84
LiEGe	18.79	44.32	<b>15.06</b>	<b>35.80</b>
LiEGe w/o MHP	18.81	44.36	14.52 $\nabla$	34.53 $\nabla$
LiEGe w/o OE	<b>18.87</b>	<b>44.39</b>	14.23 $\nabla$	34.58 $\nabla$
LiEGe w/o BA	17.55 $\nabla$	43.58 $\nabla$	13.32 $\nabla$	33.73 $\nabla$
LiEGe w/o CDA	18.33 $\nabla$	43.59 $\nabla$	14.04 $\nabla$	34.13 $\nabla$

- **LiEGe w/o CDA** does not have cross-document attention sub-layers in its decoding layers. In other words, this model incorporates cross-document interactions only during encoding.

The performance of ablated models is reported in Table 3.7. The results of models for generation of novelty explanations over Wiki-NEG show that LiEGe constantly outperforms its ablated versions and the observed improvements are statistically significant. Evaluation over Wiki-CEG for the comprehensive explanation however shows that LiEGe outperforms two of its ablated models where *broadcast & add* or *cross-document interactions* is removed. The performance differences with the other two ablated versions are not statistically significant. An on-par performance of LiEGe with the one without *ordinal document encoding* for comprehensive explanations is expected as these explanations are not dependent on the document position in a ranked list. Document representation by MHP is more important for novelty explanations compared to comprehensive ones. A possible reason for this observation is that the MHP representation of documents provides more flexibility to attend to a specific part of a document content compared to the [CLS] representation. This specific attention to a small part of a document is needed for novelty explanations, while comprehensive explanations can also be generated based on the [CLS] encoding of documents. Finally, *cross-document interactions* and *broadcast & add* are found to

be essential for both novelty and comprehensive explanations. This demonstrates the necessity and utility of listwise modeling of the SeRE task.

### 3.5 Summary

We address the problem of post-hoc interpretability of search results through content-based explanations. We introduce two new settings for explanation generation: novelty and comprehensive. To overcome the data scarcity challenge in training and evaluating these tasks, we construct weak-supervision datasets from Wikipedia and use evaluation datasets from MIMICS for both settings. Our model, LiEGe, is designed to explain all documents in a search result list by leveraging cross-document interactions within both the encoder and decoder stages of transformer-based language models. Experimental results highlight the effectiveness of LiEGe and the benefits of transfer learning from open-domain data in generating explanations, demonstrating superior performance compared to state-of-the-art baselines.

Post-hoc explanation approaches like LiEGe enhance the interpretability of search results but do not directly improve the accuracy of those results (i.e., the overall quality of SERPs). In the next chapter, we will explore how to improve both the interpretability and quality of search results simultaneously through a unified search result diversification framework.

## CHAPTER 4

# INTRINSICALLY EXPLAINABLE END-TO-END SEARCH RESULT DIVERSIFICATION

Post-hoc and model-agnostic explanations for SERPs are inherently independent of the decision-making process employed by the underlying information retrieval (IR) model that ranks the documents. In Chapter 3, we treat the query aspects contained in a document as a form of natural language explanation. Conversely, the reranking of documents based on implicit or explicit document aspects is an established task known as search result diversification (SRD), an ad-hoc retrieval task that focuses on the diversity of multiple subtopics/aspects in top-ranked positions, alongside the notion of relevance. This raises the question of whether it is feasible to generate explanations and perform SRD simultaneously.

Regarding search result diversification, there are two main categories of SRD approaches based on their diversification strategy: *coverage-based* and *novelty-based*. Coverage-based approaches concentrate on assessing how comprehensively a given document covers various aspects of the query. Conversely, novelty-based approaches compare retrieved documents against each other to promote novel information. A key advantage of coverage-based approaches over novelty-based approaches is their higher degree of model interpretability and transparency for users (MacAvaney et al., 2021), as the aspects and how documents address them are more straightforward to understand. In contrast, novelty-based approaches often rely on metrics such as dissimilarity between document embeddings (Su et al., 2021; Yan et al., 2021; Yu, 2022), which can be challenging for humans to interpret.

While coverage-based approaches are praised for their interpretability, they often depend on external systems for acquiring query aspects, such as proprietary Google query suggestions (Hu et al., 2015; Jiang et al., 2017; Qin et al., 2020, 2023) or query completion models trained with query logs (MacAvaney et al., 2021). Relying on such external systems introduces several drawbacks. First, the constant availability of these systems cannot be guaranteed due to factors such as high training or inference costs, or restrictions on extensive usage. Second, the acquisition of query aspects is not based on the actual documents to be re-ranked but relies on external query logs and click data. This disconnect means there is no assurance that the acquired query aspects are relevant to the candidate documents, potentially hindering the re-ranking process. Lastly, these systems are not optimized for the specific goals of search result diversification. Even in the case of IntenT5 (MacAvaney et al., 2021), an open-source alternative to proprietary systems, query aspects are in plain text, preventing the back-propagation of diversity-oriented losses to the query suggestion model, thereby hindering joint optimization.

We introduce DUB (short for **D**iversification **U**sing **B**ottlenecks), a coverage-based diversification framework that incorporates a differentiable aspect extraction component. This component “summarizes” relevant information from candidate documents into latent aspect embeddings, optimized to enhance diversified re-ranking. To facilitate optimization across all components, including text encoders, we adapt the differentiable clustering algorithm (Cho et al., 2021) to better handle document sets across varying levels of topic specificity. Aspect-specific representations of passages are then employed to predict query aspects. The existing SRD datasets do not offer adequate training support for our framework, presenting a data scarcity challenge. To counter this, we propose using the English Wikipedia to pre-train the more resource-intensive components of DUB on a related *explanation* task called aspect matching. By pre-training on Wikipedia and then fine-tuning on dedicated SRD

datasets, DUB demonstrates impressive performance in diversification. Compared to novelty-based approaches, DUB offers inherent explainability. Moreover, unlike other coverage-based methods, DUB eliminates reliance on external sources for query aspect acquisition, achieving a similar level of explainability.

*The work described in this chapter, namely “Search Result Diversification Using Bottlenecks”, was published in CIKM 2023 (Yu et al., 2023). I was the lead author responsible for creating datasets, designing model architectures, designing the transfer learning strategies and conducting experiments.*

## 4.1 Diversification Using Bottlenecks (DUB)

In this section, we detail our proposed framework for end-to-end explainable search result diversification. We begin by outlining the diversification task, followed by an introduction to the first component of our framework, the text encoder. Next, we discuss the core component, the neural aspect extractor, and present two distinct methods for constructing it. Finally, we briefly explain how the diversified ranker leverages aspects identified by the aspect extractor to diversify document rankings.

### 4.1.1 Task Formulation and Model Overview

Consider a search query denoted as  $q$  and a ranked list of candidate documents represented as  $R$ . An SRD model, denoted as  $\mathcal{F}$ , generates a list of ranking scores,  $S = \mathcal{F}(q, R)$ . The goal is to obtain a re-ranked list  $\pi(R)$  according to  $S$ , which is expected to exhibit higher diversity compared to the original ranked list  $R$  without compromising relevance.

The DUB framework  $\mathcal{F}$  in our study consists of three learnable components, as depicted in Figure 4.1: the text encoder  $\mathcal{E}$ , the aspect extractor  $\mathcal{A}$ , and the diversified ranker  $\mathcal{P}$ . The text encoder  $\mathcal{E}$  is responsible for obtaining the query embedding  $\mathbf{q}$  and passage embeddings  $\mathbf{P}$ . These embeddings are further utilized to select candi-

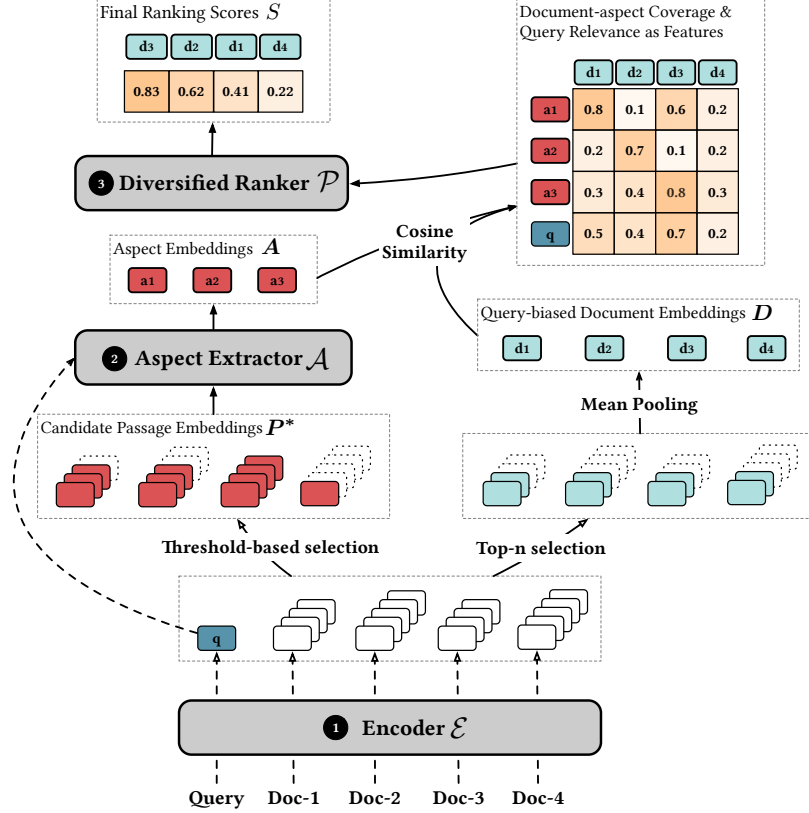


Figure 4.1: An overview of the DUB framework.

date passage embeddings  $P^*$  and to build query-biased document embeddings  $D$ . Subsequently, the aspect extractor  $\mathcal{A}$  leverages the candidate passages embeddings  $P^*$  and query embeddings  $q$  to produce aspect embeddings  $A$ , which serve as information bottlenecks for the diversification task. Lastly, the diversified ranker  $\mathcal{P}$  takes the aspect embeddings  $A$  and the query-biased document embeddings  $D$ , calculates document-aspect coverage as document features, and outputs scalar ranking scores  $S$ . We now introduce each component in detail.

#### 4.1.2 Text Encoder

Recent works on search result diversification (Jiang et al., 2017; Qin et al., 2020, 2023; Su et al., 2021; Yan et al., 2021; Yu, 2022) often utilize unsupervised Doc2Vec embeddings (Le and Mikolov, 2014) to represent queries, aspects, and documents.

However, we draw inspiration from the successful application of contextualized representations in various NLP and IR tasks and employ a shared Transformer-based language model as both the query and document encoder. This allows DUB to be optimized end-to-end with respect to *input texts*.

To accommodate the length limitations of typical encoders, such as the 512-token limit of BERT (Devlin et al., 2019), and to address the relevance and multi-aspect coverage of long documents, we segment documents into overlapping passages for encoding. This strategy is based on three considerations: (1) the length of documents often exceeds the input capacity of the encoder; (2) only a portion of a long document may be relevant to the query (TREC, 2000); and (3) a single document may cover multiple query aspects, potentially leading to information loss if represented by a single embedding (Luan et al., 2021). We use the mean of the token embeddings from the last encoder layer to obtain the query embedding ( $\mathbf{q}$ ) and passage embeddings ( $\mathbf{P}$ ).

After encoding, we filter out less relevant passages and derive two types of embeddings: candidate passage embeddings for aspect extraction and query-biased document embeddings for document scoring. For candidate passage embeddings, passages with a cosine similarity to the query greater than a preset threshold  $\theta$  are selected, represented as  $\mathbf{P}^* = \{\mathbf{p} \mid \cos(\mathbf{q}, \mathbf{p}) \geq \theta, \mathbf{p} \in \mathbf{P}\}$ , used for aspect extraction. For query-biased document embeddings, we select the top- $n$  most similar passages to the query from each document and average these to form a query-biased embedding, denoted by  $\mathbf{d}$ . The collection of these embeddings across all candidate documents is denoted as  $\mathbf{D}$ .

This dual selection approach aims to filter out irrelevant content and is particularly effective for handling candidate documents that may lack suitable passages for creating an informative query-biased document embedding if solely relying on threshold-based selection.

### 4.1.3 Neural Aspect Extractor

After obtaining the query embedding  $\mathbf{q}$  and candidate passage embeddings  $\mathbf{P}^*$ , the aspect extractor  $\mathcal{A}$  is employed to generate a fixed number ( $K$ ) of aspect embeddings per query  $\mathbf{A} = \mathcal{A}(\mathbf{q}, \mathbf{P}^*)$ , where  $\mathbf{A} \in \mathbb{R}^{K \times d}$ . Each aspect embedding is designed to capture a specific aspect of the query-relevant information covered by the retrieved documents. We explore two different methods for extracting query aspects.

#### 4.1.3.1 Aspect Extractor Using Multi-Head Attention

The first design for the DUB aspect extractor utilizes multi-head attention (MHA) to derive query aspects from similar passages. We introduce a modification to the original MHA framework (Vaswani et al., 2017) and its implementation in aspect-based dense retrieval (Kong et al., 2022). Specifically, we treat the output of each attention head as the latent representation of a query aspect, similar to the intent modeling approach proposed by Chen et al. (2020a). Unlike typical implementations where the outputs of  $h = K$  attention heads are combined, we preserve these outputs separately as  $K$  distinct aspect embeddings. The formal implementation of  $\mathcal{A}$  is as follows:

$$\mathbf{A} = \mathcal{A}(\mathbf{q}, \mathbf{P}^*) = \text{MHA}(\mathbf{q}, \mathbf{P}^*, \mathbf{P}^*) = \{\text{head}_1, \dots, \text{head}_K\}, \quad (4.1)$$

$$\mathbf{a}_i = \text{head}_i = \text{Attn}(\mathbf{q}\mathbf{W}_i^Q, \mathbf{P}^*\mathbf{W}_i^K, \mathbf{P}^*\mathbf{W}_i^V), \quad (4.2)$$

Here, the query embedding  $\mathbf{q}$  functions as the query matrix in the self-attention mechanism, with the candidate passage embeddings  $\mathbf{P}^*$  acting as both key and value matrices.

Notably, the input projection matrices  $\mathbf{W}_i^Q$ ,  $\mathbf{W}_i^K$ , and  $\mathbf{W}_i^V$  in DUB’s aspect extractor are dimensioned as  $\mathbb{R}^{d \times d}$ , differing from the matrices in traditional models,

which are dimensioned as  $\mathbb{R}^{d \times d/h}$ . This modification ensures that the output of each head has the necessary dimensionality of  $d$  to effectively represent one query aspect.

#### 4.1.3.2 Aspect Extractor Using GDKM-based Clustering

An alternative intuition for query aspect extraction is the premise that passages covering the same query aspect typically have more similar embeddings compared to those covering different aspects (Su et al., 2021). Building on this concept, we utilize clustering on candidate passage embeddings  $\mathbf{P}^*$  to derive aspect representations.

In the clustering-based aspect extractor component, we directly incorporate passage interactions, unlike the MHA-based approach, which captures these interactions indirectly through their similarity to the query embedding. The clustering-based aspect extraction process consists of two main steps: (1) clustering the passage embeddings to group similar content; and (2) generating an aspect embedding from the passages within each cluster. This method allows for a more direct analysis of the relationships among passages, facilitating more distinct and coherent aspect identification.

*Step 1: Clustering passages with Generalized Differentiable K-means.* The clustering component for aspect extraction must be differentiable so that the encoder ( $\mathcal{E}$ ) parameters can be optimized using gradients from the loss function. To overcome the limitations<sup>1</sup> of DKM (Cho et al., 2021) for query-specific passage clustering, we introduce GDKM, a generalization of DKM. This approach limits the number of clusters

---

<sup>1</sup>DKM achieves differentiability through an attention-based soft assignment, allowing each instance to belong to *all* clusters with varying attention weights. However, DKM can converge to a trivial solution where every instance forms the same  $K$  clusters, which is undesirable. Cho et al. (2021) attempted to mitigate this by limiting the process to a maximum of five clustering iterations to prevent such convergence. While this may be effective for clustering stable sets of instances, it is not adaptable for dynamically changing sets of passages across different queries. The optimal number of clustering iterations might vary by query.

---

**Algorithm 1:** GDKM algorithm

---

**Input:** Passage embeddings  $\mathbf{P}^*$ , minimum moving distance  $\epsilon$ , number of clusters  $K$ , temperature  $\tau$ , degree of freedom  $\nu$ , and mask attention value  $\iota$ .

**Output:** Cluster assignments  $\tilde{\alpha}$  and centroids  $\tilde{\mu}$

```
1 Function GDKM( $\mathbf{P}^*$ ,  $\epsilon$ ,  $K$ ,  $\tau$ ,  $\nu$ ,  $\iota$ ):
2    $\mu \leftarrow K$ -means++( $\mathbf{P}^*$ ,  $K$ )           // Initialization;  $|\mu| == K$ 
3   while True do
4      $\delta \leftarrow \{\delta_{ij} = \cos(\mathbf{p}_i, \mu_j)\}$ ,  $1 \leq i \leq |\mathbf{P}^*|, 1 \leq j \leq |\mu|$ 
5      $\alpha \leftarrow \{\alpha_{ij} = \frac{\exp(\delta_{ij}/\tau)}{\sum_j \exp(\delta_{ij}/\tau)}\}$ ,  $1 \leq i \leq |\mathbf{P}^*|, 1 \leq j \leq |\mu|$ 
6     for  $i = 1, 2, \dots, |\mathbf{P}^*|$  do
7        $t \leftarrow \text{sort-desc}(\alpha[i])[\nu]$            //  $\nu$ -largest in  $\alpha[i]$ 
8       for  $j = 1, 2, \dots, |\mu|$  do
9         if  $\alpha_{ij} \geq t$  then
10           $\tilde{\alpha}_{ij} \leftarrow \alpha_{ij}$ 
11        else
12           $\tilde{\alpha}_{ij} \leftarrow \iota$ 
13        end
14      end
15    end
16     $\tilde{\mu} \leftarrow \{\tilde{\mu}_j = \frac{\sum_i \tilde{\alpha}_{ij} \mathbf{p}_i}{\sum_i \tilde{\alpha}_{ij}}\}$ ,  $1 \leq i \leq |\mathbf{P}^*|, 1 \leq j \leq |\mu|$ 
17    if  $\|\tilde{\mu} - \mu\| \leq \epsilon$  then
18       $\tilde{\alpha} \leftarrow \{\tilde{\alpha}_{ij}\}$ ,  $1 \leq i \leq |\mathbf{P}^*|, 1 \leq j \leq |\mu|$ 
19      return  $\tilde{\alpha}, \tilde{\mu}$            // converge and exit
20    else
21       $\mu \leftarrow \tilde{\mu}$            // go to the next iteration
22    end
23  end
```

---

to which each instance (passage) can be assigned, allowing a passage to belong to multiple, but *not all*, clusters in a probabilistic manner.

We introduce a hyperparameter, *degree of freedom*, denoted by  $\nu$ , which specifies the maximum number of clusters an instance can belong to. GDKM offers a flexible framework as it can mimic the original  $K$ -means algorithm (MacQueen, 1967) when  $\nu$  is set to 1, and can replicate DKM when  $\nu$  is set to  $K$ . By setting  $\nu$  to an integer between 1 and  $K$ , GDKM effectively models passages covering multiple aspects of the query without encountering the convergence issues typical of DKM. The broader applicability of GDKM for other tasks is a subject for future exploration.

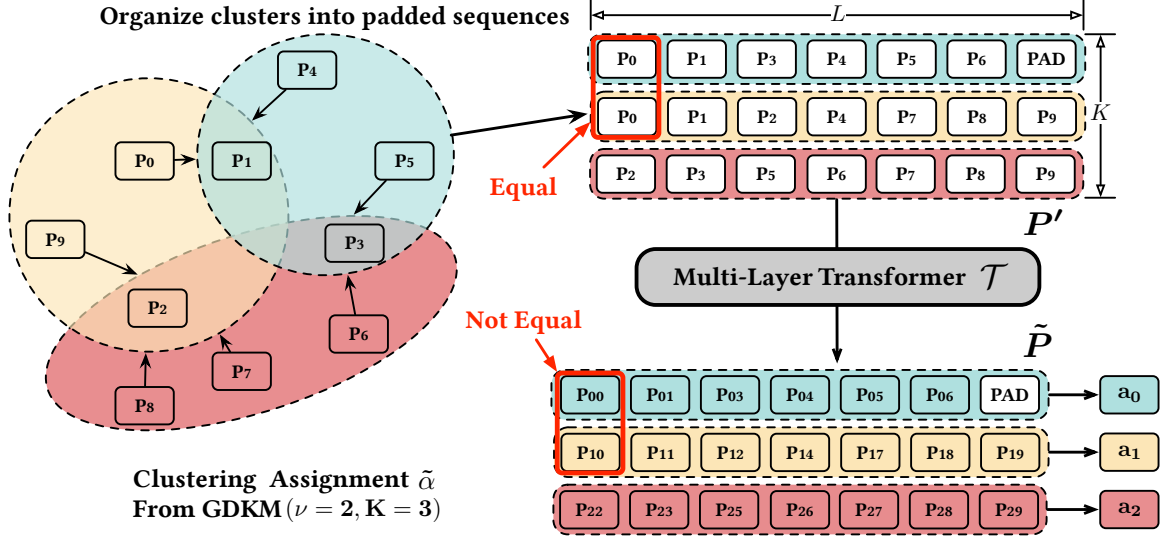


Figure 4.2: Aspect extraction using GDKM clustering ( $K=3, \nu=2$ ). Each cluster contains passages that are inside its border and passages that point to its border. E.g., both  $p_0$  and  $p_4$  are part of the yellow cluster, with the former having a higher probability, whereas  $p_3$  does not belong to this cluster.

The pseudocode of GDKM is presented in Algorithm 1. Highlighted lines indicate the extension to DKM. For each passage embedding, the clustering layer first estimates the probability of its membership in each latent cluster, resulting in an attention matrix denoted by  $\alpha$  in line 5. Instead of using this attention matrix to compute new centroids as in DKM, we only keep the highest  $\nu$  attention weights per passage (line 10) and mask the rest with a small constant value  $\iota$  (line 12). New cluster centroids are then calculated based on the masked attention matrix  $\tilde{\alpha}$ . After convergence, GDKM outputs cluster assignments  $\tilde{\alpha}$  and cluster centroids  $\tilde{\mu}$  from the final iteration.

*Step 2: Generating aspect embeddings from clusters.* A passage that belongs to at most  $\nu$  clusters can be represented in up to  $\nu$  distinct ways. In DUB, each passage is therefore represented with  $\nu$  embeddings, each corresponding to a potential query aspect it might cover. To facilitate this, DUB employs a multi-layer Transformer, denoted by  $\mathcal{T}$ , to derive aspect-specific representations of passages  $\tilde{P}$  from the initial

embeddings  $\mathbf{P}^*$  and their cluster assignments  $\tilde{\alpha}$ , such that  $\tilde{\mathbf{P}}, \alpha' = \mathcal{T}(\mathbf{P}^*, \tilde{\alpha})$ . This process is depicted in Figure 4.2.

We use the clustering assignment  $\tilde{\alpha}$  to organize  $\mathbf{P}^*$  into  $K$  sequences of passage embeddings  $\mathbf{P}'$ , where each passage embedding from  $\mathbf{P}^*$  is replicated  $\nu$  times. We pad shorter sequences with zero vectors for batching and denote the padded sequence length as  $L$ . The clustering assignment  $\tilde{\alpha}$  is reformatted into  $\alpha'$  by removing entries with masked value  $\iota$  (as these do not appear in  $\mathbf{P}'$ ) and adding entries for padded embeddings with  $\iota$ , resulting in  $\alpha' \in \mathbb{R}^{K \times L}$ .

The multi-layer Transformer  $\mathcal{T}$  applies in-sequence (in-cluster) self-attention to update the passage embeddings, allowing each passage to be influenced by others in the same cluster that cover the same query aspect. This refined, aspect-specific passage embedding approach,  $\tilde{\mathbf{P}}$ , reduces ambiguity compared to  $\mathbf{P}^*$  and enhances the accuracy of the aspect embeddings. Finally, the aspect embeddings  $\mathbf{A}$  are calculated by averaging the aspect-specific embeddings  $\tilde{\mathbf{P}} \in \mathbb{R}^{K \times L \times d}$ , weighted by their degree of membership  $\alpha' \in \mathbb{R}^{K \times L}$ , expressed as  $\mathbf{A} = \text{Softmax}(\alpha') \tilde{\mathbf{P}}^\top$ , where  $\tilde{\mathbf{P}}^\top$  signifies the transposition of the first two dimensions of  $\tilde{\mathbf{P}}$ .

#### 4.1.4 Diversified Ranker

Explicitly modeling the possible query aspects covered by candidate documents allows us to estimate their relevance to those aspects and provide a diversified ranking of retrieved results, similar to explicit models (Dang and Croft, 2013, 2012; Hu et al., 2015; Jiang et al., 2017; Qin et al., 2020, 2023; Santos et al., 2010a, 2012; Sarwar et al., 2020). For this purpose, we first estimate document-aspect coverage by simply taking the cosine similarity of query-biased document embeddings  $\mathbf{D}$  and aspect embeddings  $\mathbf{A}$ . In addition, we also use the cosine similarity between document embeddings  $\mathbf{D}$  and the original query embedding  $\mathbf{q}$  to represent documents' overall relevance to the query. Thus, a document is represented with  $K + 1$  features.

In the *score-and-sort* re-ranking paradigm, we employ a multi-layer feed-forward neural network  $\mathcal{P}$  with batch normalization. This approach follows previous diversification studies (Yan et al., 2021; Yu, 2022) and serves as the mechanism for our diversified ranker.

The formal definition of the diversified ranker  $\mathcal{P}$  can then be represented as follows:

$$S = \mathcal{P}(\text{Concat}(\cos(\mathbf{D}, \mathbf{A}); \cos(\mathbf{D}, \mathbf{q}))), \quad (4.3)$$

## 4.2 Addressing Data Scarcity with Explanation-based Pre-training

DUB comprises three learnable components  $\mathcal{F} = \{\mathcal{E}, \mathcal{A}, \mathcal{P}\}$ , designed to be trained end-to-end on a search result diversification dataset. This training approach allows the text encoder  $\mathcal{E}$  and the aspect extractor  $\mathcal{A}$  to be optimized specifically for the task of diversified re-ranking. However, a significant challenge in training DUB is the scarcity of data, particularly evident in the largest publicly accessible SRD dataset, TREC Web Tracks, which contains fewer than 200 queries in total from year 2009 to 2012. To mitigate this data scarcity, we propose a strategy that involves pre-training the parameter-intensive components  $\{\mathcal{E}, \mathcal{A}\}$  on a related explanation task, which has access to a larger volume of weak-supervision data. Following this pre-training phase, the entire model  $\mathcal{F} = \{\mathcal{E}, \mathcal{A}, \mathcal{P}\}$  undergoes end-to-end fine-tuning for SRD, enabling comprehensive optimization and application to the diversification task.

### 4.2.1 Pre-training with Aspect Matching

Inspired by previous research that utilizes the structured data from Wikipedia to simulate the query-aspect-passage structure typical in information retrieval tasks (Dietz et al., 2017; Rahimi et al., 2021) and our previous approach (Section 3.1.1), we generate weak training data for initial pre-training. This prepares DUB’s components to better extract aspects from free text.

#### 4.2.1.1 Aspect Pre-training Data

For the pre-training of the text encoder  $\mathcal{E}$  and aspect extractor  $\mathcal{A}$ , we create a weakly supervised dataset from Wikipedia, termed **Wiki**. This dataset utilizes Wikipedia articles, using their titles as search queries, their section headings as query aspects, and the content sections (excluding the introduction) as multiple relevant passages. We select articles with eight or more sections, resulting in 203,751 training samples. During pre-training, DUB randomly samples  $K$  aspects per query, and reference aspect embeddings  $\mathbf{A}_r$  are created by encoding the concatenation of the article title (query) and section headings (aspects).

#### 4.2.1.2 Aspect Matching Task

Each training sample consists of a query  $q$ ,  $K$  reference aspects  $A_r$ , and passages relevant to both the query and these aspects. The embeddings of these reference aspects are generated by  $\mathbf{A}_r = \mathcal{E}(A_r)$ , where  $\mathbf{A}_r \in \mathbb{R}^{K \times d}$ . The primary goal of this pre-training task is to align the predicted aspects ( $\mathbf{A}$ ) with the reference aspects ( $\mathbf{A}_r$ ) within the embedding space. Given the inherent difficulty in directly minimizing their pairwise differences due to lack of clear alignments, we introduce two innovative solutions.

#### 4.2.1.3 Optimal-Transport Based Objective

We cast the alignment of predicted and reference aspect embeddings as an instance of the optimal transport (OT) problem and solve it with an existing OT solver. This is inspired by works on aligning token embeddings from different languages (Alqahtani et al., 2021; Huang et al., 2023b; Nguyen and Luu, 2022). In this OT formulation, we define the cost matrix  $\mathbf{M}$  as the pairwise cosine distance between predicted aspect embeddings  $\mathbf{A}$  and reference aspect embeddings  $\mathbf{A}_r$ . The error in matching these two sets of embeddings is the total *transportation cost* from  $\mathbf{A}$  to  $\mathbf{A}_r$ , defined as  $\boldsymbol{\gamma} \cdot \mathbf{M}$ , where  $\boldsymbol{\gamma} \in \mathbb{R}^{K \times K}$  is called the *transportation matrix*. The  $\boldsymbol{\gamma}_*$  that minimizes the

transportation cost is called the optimal transport matrix, which intuitively represents the optimal alignment between  $\mathbf{A}$  and  $\mathbf{A}_r$ . To overcome the intractability of the linear programming solutions for finding  $\gamma_*$ , we use the IPOT algorithm (Xie et al., 2020) to compute the OT matrix  $\gamma_*$ . Finally, we define the objective for aspect matching as the optimal transportation cost:

$$\mathcal{L}_{\text{OT}}(\mathbf{A}, \mathbf{A}_r) = \gamma_* \cdot \mathbf{M}, \quad (4.4)$$

#### 4.2.1.4 Teacher-Forcing Based Objective

The clustering-based aspect extractor can be trained with an alternate objective. Note that this aspect extractor comprises an GDKM clustering layer (without trainable parameters) and a multi-layer Transformer  $\mathcal{T}$  (with trainable parameters). Only GDKM causes nondeterministic matching between predicted and reference aspect embeddings. Therefore, during the pre-training step, we can skip GDKM and directly give true ‘‘clustering’’ assignment  $\hat{\alpha}$  as input of  $\mathcal{T}$ . This is similar to teacher-forcing training (Williams and Zipser, 1989) used for training sequence generation models (Raffel et al., 2020). This provides training stability by eliminating potential misalignment of embedding sets from the OT solver. The loss function of this training method is defined based on the cosine distance of matching embeddings as:

$$\mathcal{L}_{\text{TF}}(\mathbf{A}, \mathbf{A}_r) = \sum_{i=1}^K (1 - \cos(\mathbf{a}, \mathbf{a}_i^r)), \quad (4.5)$$

We observe slightly better performance of DUB-GDKM using this loss for pre-training compared with the OT-based loss (Eq. 4.4).

#### 4.2.2 End-to-end SRD Training

A multi-layer feed-forward neural network  $\mathcal{P}$  predicts ranking scores  $S$  for documents in  $R$  based on  $(K + 1)$ -dimensional features of aspect coverage and query

similarity. We use the  $\alpha$ -DCG loss (Yan et al., 2021) to optimize the entire DUB model  $\mathcal{F} = \{\mathcal{E}, \mathcal{A}, \mathcal{P}\}$  using training data with aspect-level relevance judgements.

### 4.3 Experimental Setup

In this section, we detail the datasets, baseline approaches, and metrics used to evaluate the search result diversification performance of DUB.

#### 4.3.1 Evaluation Datasets

**TREC-Web.** The TREC Web track datasets from 2009 to 2012 are used for evaluating search result diversification (Jiang et al., 2017; Qin et al., 2020, 2023; Su et al., 2021; Yan et al., 2021). The combined dataset, referred to as **TREC-Web**, consists of 198 topics after excluding two topics without subtopic judgments, and documents from the ClueWeb’09-Category B collection. Five-fold cross-validation is conducted using the same data folds as previous works (Jiang et al., 2017; Qin et al., 2023).

**MIMICS-Div.** The **MIMICS-Div** dataset is constructed based on the “Click-Explore” version of the MIMICS datasets (Zamani et al., 2020). We repurpose MIMICS to evaluate search result diversification, particularly to simulate a scenario with abundant real queries and investigate the effects of pre-training on large-scale open-domain data. Specifically, each candidate answer for a query-clarification pair is considered as a query aspect. The MIMICS datasets do not provide full document contents or relevance labels. Following prior study (Hashemi et al., 2021), we consider the concatenation of a document’s heading and snippet as its content, and a document is deemed relevant to a query aspect if it contains all the aspect terms. MIMICS-Div contains 8,166 queries with an average of 3.17 aspects per query. It is important to note that the relevance assessments in MIMICS-Div are not manually verified, but automatically inferred based on terms overlaps (see Section 3.1.2).

### 4.3.2 Competing Methods

In this section, we categorize competing search result diversification methods into three groups and provide brief explanations of their similarities to and discrepancies from our method, DUB.

**(1) Explicit models by extracting aspects from search results:** We develop two baselines based on topic modeling (Carterette and Chandar, 2009), which align with DUB’s approach of explicit diversification through aspect extraction from candidate documents. These baselines use unsupervised methods to extract latent topics (representing query aspects in our context) and model each topic as a probability distribution over the vocabulary (an unigram language model). The probability of a document covering a query aspect,  $\Pr(d_i|a_j)$ , is approximated by  $\prod \Pr(v|a_j)$  for each document and aspect. We employ the unsupervised explicit diversification algorithm xQuAD (Santos et al., 2010b) to re-rank candidate documents. In xQuAD, we use uniform aspect importance distributions and tune the balance parameter  $\lambda$  (for balancing relevance and diversity) through cross-validation. We compute aspect models using LDA (Blei et al., 2003) and the neural topic model BERTopic (Grootendorst, 2022), resulting in the baselines LDA-xQuAD and BERTopic-xQuAD.

**(2) Neural implicit models:** We focus on two recent implicit SRD models, namely Graph4Div (Su et al., 2021) and DALETOR (Yan et al., 2021). These models, like DUB, do not depend on predefined query aspects and instead infer relevant aspects implicitly.

**(3) Explicit models utilizing aspects from external sources:** This category includes unsupervised explicit SRD methods such as xQuAD (Santos et al., 2010a), PM2 (Dang and Croft, 2012), and HxQuAD/HPM2 (Hu et al., 2015), which rely on query aspects derived from external sources. Additionally, DSSA (Jiang et al., 2017), DESA (Qin et al., 2020), and GDESA (Qin et al., 2023) utilize Google’s first-level

query suggestions as external aspects but differ as they are supervised neural models optimized with aspect-level judgments.

### 4.3.3 Evaluation Metrics

We adopt the official TREC evaluation methodology for the diversity task. On TREC-Web, we report the following evaluation metrics with a cut-off set to 20, as done in previous studies:  $\alpha$ -nDCG, ERR-IA, NRBP, Pre-IA, and S-rec. We set the parameter  $\alpha$  to 0.5, the default setting in the official TREC evaluation program. On MIMICS-Div, we report  $\alpha$ -nDCG@{5,10} and ERR-IA@{5,10} since the candidate set contains at most 10 documents. For conducting statistical significance tests, we employ the  $t$ -test with Bonferroni correction at the 95% confidence level. Statistical improvements that are significant over all baseline models are indicated with a † symbol in the result tables.

## 4.4 Experimental Results and Analysis

The performance of various approaches on the TREC-Web dataset is detailed in Table 4.1, where results are grouped based on the representation of queries and documents. Baselines are categorized according to the characteristics outlined in Section 4.3.2. Notably, DUB-GDKM outperforms all baseline groups, underscoring its efficacy. The improvements are statistically significant across all metrics, with the exception of Pre-IA. Below, we further analyze key observations from the results.

### 4.4.1 Importance of Supervised Aspect Extraction

The results in Table 4.1 indicate that all explicit baselines utilizing aspects from external sources (category (3)) outperform those that extract aspects from top-retrieved documents using topic modeling (category (1)). Despite the prevailing trend in prior studies, our DUB, which extracts aspects from top-retrieved documents, surpasses the performance of supervised explicit models (DSSA, DESA, and GDESA) that uti-

Table 4.1: Search result diversification on TREC-Web.

#	Metric	$\alpha$ -nDCG	ERR-IA	NRBP	Pre-IA	S-rec
<b>Term-level Representations</b>						
1	(1) LDA-xQuAD	0.335	0.224	0.183	0.127	0.608
2	(3) GQS-xQuAD	0.413	0.317	0.284	0.161	0.622
3	(3) GQS-PM2	0.411	0.306	0.267	0.169	0.643
4	(3) GQS-HxQuAD	0.421	0.326	0.294	0.158	0.629
5	(3) GQS-HPM2	0.420	0.317	0.279	0.172	0.645
<b>SBERT as Text Encoder</b>						
6	(1) BERTopic-xQuAD	0.330	0.232	0.199	0.140	0.555
7	(2) DALETOR	0.411	0.317	0.278	0.151	0.614
8	(2) Graph4Div	0.475	0.375	0.343	0.187	0.669
9	(3) DSSA	0.461	0.357	0.324	0.185	0.649
10	(3) DESA	0.473	0.370	0.338	0.185	0.657
11	(3) GDESA	0.478	0.376	0.344	0.186	0.666
12	DUB-MHA	0.497 <sup>†</sup>	0.391 <sup>†</sup>	0.363 <sup>†</sup>	0.188	0.674
13	DUB-GDKM	<b>0.508<sup>†</sup></b>	<b>0.399<sup>†</sup></b>	<b>0.374<sup>†</sup></b>	<b>0.190</b>	<b>0.680<sup>†</sup></b>
<b>Ablations</b>						
14	DUB-MHA (no-PRE)	0.473	0.372	0.336	0.185	0.663
15	DUB-GDKM (no-PRE, $\nu=2$ )	0.461	0.360	0.320	0.184	0.658
16	DUB-GDKM ( $\nu=1$ )	0.493 <sup>†</sup>	0.387 <sup>†</sup>	0.358 <sup>†</sup>	0.188	0.673
17	DUB-GDKM ( $\nu=3$ )	0.506 <sup>†</sup>	0.397 <sup>†</sup>	0.371 <sup>†</sup>	<b>0.190</b>	0.679 <sup>†</sup>
18	DUB-GDKM ( $\nu=8$ )	0.437	0.343	0.293	0.181	0.647

lize Google query suggestions as aspects (#12-13 vs. #9-11). This finding highlights that an effective aspect extractor, capable of deriving query aspects from top-retrieved documents, can significantly enhance search result diversification, offering advantages over reliance on external sources for aspect generation based solely on the query.

#### 4.4.2 Utility of Pre-training

As shown in Table 4.1, variants of DUB that were not pre-trained on Wiki (indicated as “no-PRE”) display reduced performance on TREC-Web (#14-15 vs. #12-13). This decline in effectiveness is particularly notable in the GDKM-based model compared to the MHA-based model, which can be attributed to the larger parameter count of the GDKM-based model, necessitating more extensive training data.

Table 4.2: Search result diversification on MIMICS-Div.

<b>Metric</b>	$\alpha$ -nDCG@5	$\alpha$ -nDCG@10	ERR-IA@5	ERR-IA@10
Graph4Div	0.614	0.701	0.420	0.459
DALETOR	0.678	0.759	0.484	0.512
MO4Div	0.679	0.762	0.487	0.515
DUB-MHA	0.699 <sup>†</sup>	0.790 <sup>†</sup>	0.512 <sup>†</sup>	0.543 <sup>†</sup>
DUB-GDKM	<b>0.705<sup>†</sup></b>	<b>0.797<sup>†</sup></b>	<b>0.514<sup>†</sup></b>	<b>0.549<sup>†</sup></b>
<b>Ablations</b>				
DUB-MHA (no-PRE)	0.687 <sup>†</sup>	0.780 <sup>†</sup>	0.503 <sup>†</sup>	0.532 <sup>†</sup>
DUB-GDKM (no-PRE)	0.682 <sup>†</sup>	0.773 <sup>†</sup>	0.498 <sup>†</sup>	0.524 <sup>†</sup>

#### 4.4.3 Diversification on MIMICS-Div

For a comprehensive analysis of the role of pre-training, we evaluated the performance of our models and applicable baselines on the MIMICS-Div dataset, which, unlike TREC-Web, contains numerous training queries. The results are presented in Table 4.2. We observe that DUB significantly outperforms strong baselines, even without pre-training on Wiki. This suggests that pre-training on Wiki is not crucial for DUB when a sufficient number of search result diversification training queries are available. However, the additional improvements seen with pre-training on Wiki still underscore the value of this transfer learning strategy.

#### 4.4.4 Discussion: MHA vs. GDKM

In this section, we discuss the strengths and weaknesses of the two implementations of the aspect extraction component  $\mathcal{A}$ . DUB-GDKM delivers the best diversification performance on both evaluation datasets and provides a higher degree of *attributability*, enabling easier tracking of an extracted aspect back to a cluster of passages. Conversely, DUB-MHA, while less effective than DUB-GDKM, still surpasses all baseline models. It benefits from having fewer parameters and not utilizing a clustering layer, offering two key advantages: (1) DUB-MHA requires less training data and performs better without pre-training, as evidenced by the results in Tables 4.1 and 4.2; and (2) DUB-MHA is more efficient in processing.

## 4.5 Evaluation of Latent Aspects

In this section, we directly assess the efficacy of latent aspects (embeddings) generated by DUB, specifically examining their diversity and relevance not just for diversified reranking.

### 4.5.1 Compared Methods

We evaluate the aspect representations generated by LDA, BERTopic, Google query suggestions (referred to as GQS) (Hu et al., 2015), GPT-3.5 (Brown et al., 2020), and aspect embeddings from DUB-GDKM on the TREC-Web dataset. For GQS and GPT, we consider the first 8 aspects per query. Notably, aspects from LDA, GQS, and GPT are presented in textual form, while aspects from BERTopic and DUB-GDKM are represented using SBERT embeddings. To facilitate comparisons, we use the same encoder  $\mathcal{E}$  to transform GQS and GPT query aspects into aspect embeddings. During this transformation, we exclude embeddings corresponding to the query terms to prevent the aspect embeddings from becoming overly similar and thus reducing diversity scores.

Additionally, we convert the aspect embeddings from BERTopic and DUB-GDKM into tokens by selecting the top-5 tokens whose embeddings from  $\mathcal{E}$  most closely match each aspect embedding. This method is consistent with established practices for interpreting Transformer embeddings (Dar et al., 2022; Geva et al., 2021). We apply the same tokenization approach to GQS and GPT aspect embeddings, ensuring that each aspect is adequately expanded and represented by 5 tokens (even though many originally have just 1 token per aspect).

This dual approach allows us to compare aspects both in their textual form and their latent representations, offering a comprehensive view of their utility.

Table 4.3: Evaluating the quality of extracted aspects.

Metric	Diversity		Relevance	
	token	embedding	$\Delta$ -MAP	$\Delta$ -nDCG
RM3	-	-	+0.020	+0.011
LDA	0.287	-	-0.007	-0.005
BERTopic	0.661	0.272	+0.006	+0.004
GQS	0.887	0.533	+0.008	+0.005
GPT	<b>0.917</b>	<b>0.673</b>	+0.003	+ 0.001
DUB-GDKM	0.862	0.404	<b>+0.026<sup>†</sup></b>	<b>+0.017<sup>†</sup></b>

#### 4.5.2 Measuring Diversity

We assess the diversity of aspects using two distinct metrics. Firstly, we calculate the average dissimilarity of aspect embeddings—this involves computing the dissimilarity for all pairs of aspects per query and then averaging these values across all queries. We utilize cosine *distance* to measure the dissimilarity between embeddings.

Secondly, we measure token diversity, which quantifies the percentage of unique tokens within the top 5 tokens of all aspect models. This metric, originally proposed to assess the diversity of topic models (Dieng et al., 2020), provides insight into the variety of terms generated by each model.

Table 4.3 presents the diversity scores of various aspect models. At both the token and embedding levels, DUB-GDKM surpasses traditional topic model baselines such as LDA and BERTopic. However, GPT generates the most diverse aspects overall. It is important to note that DUB-GDKM is configured to extract 8 aspects per query, despite many queries, according to TREC labels, typically featuring fewer than 8 distinct aspects. This configuration might lead to some overlap among the aspects produced by DUB-GDKM. Conversely, aspects derived from GQS and GPT usually encompass a broader range of unique query intents, although there is no assurance of their relevance to any specific document in the corpus.

### 4.5.3 Measuring Relevance

Query expansion with language modeling (Ponte and Croft, 1998) serves as an extrinsic evaluation of aspect models. The underlying intuition is that better quality extracted aspects lead to enhanced retrieval performance when these aspects are used to expand the original query. The expanded language model for the query is formulated as follows:

$$\Pr_+(t|q) = \beta \Pr_{\text{ML}}(t|q) + (1 - \beta) \Pr(t|\mathbf{A}), \quad (4.6)$$

where  $\Pr_{\text{ML}}(t|q)$  is the maximum-likelihood language model of the original query,  $\Pr(t|\mathbf{A})$  is the language model derived from the aspects, and  $\beta$  is a tuning hyperparameter. Aspect language models are computed by:

$$\Pr(t|\mathbf{A}) = \frac{\sum_{i=1}^8 \Pr(t|a_i)}{\sum_{t \in V} \sum_{i=1}^8 \Pr(t|a_i)}. \quad (4.7)$$

For the query expansion, we incorporate the top 40 terms (5 terms for each of the 8 aspects) from each aspect model, using the top 50 documents as the source for aspect extraction (except for GQS and GPT, which do not require documents). We also compare these results with RM3 (Abdul-Jaleel et al., 2004; Lavrenko and Croft, 2017), a robust query expansion baseline. RM3 settings include selecting 40 expansion terms from the top 50 documents, and the interpolation parameter  $\beta$  is set to its default value of 0.6. The retrieval index is constructed using all documents from TREC-Web, not the entire ClueWeb’09 Category B. In Table 4.3, we report the *performance difference* between using the maximum-likelihood estimate of the query language model and the expanded query language model, measured by MAP and nDCG, denoted as  $\Delta$ -MAP and  $\Delta$ -nDCG, respectively. Notably, the aspect models

derived from DUB-GDKM significantly outperform all other methods. This indicates that DUB-GDKM is capable of generating query aspects that are most helpful in retrieving relevant documents, offering an explanation for its superior effectiveness in diversification.

## 4.6 Summary

We introduce DUB, an interpretable-by-design ad-hoc ranking model for search result diversification that utilizes latent query aspects extracted from candidate documents as rationales. This end-to-end learnable framework not only preserves the intrinsic interpretability of coverage-based SRD frameworks but also maximizes effectiveness. It incorporates latent aspect embeddings to facilitate the joint optimization of a text encoder, a query aspect extractor, and a diversified document ranker. Additionally, DUB enhances its capabilities by optionally leveraging knowledge from Wikipedia through pre-training, effectively addressing the challenge of data scarcity in search result diversification. Experimental results show that DUB significantly outperforms existing state-of-the-art diversification models.

So far, we have employed Transformer-based language models to generate aspects as explanations of relevance (Chapter 3) and to learn contextualized, fine-grained query aspect representations within a highly effective search result diversification framework (Chapter 4). The queries targeted so far are restricted to being ambiguous or underspecified, with explanations that are aspect-like in nature. In the remaining chapters, we aim to broaden the scope of natural language explanations beyond mere query aspects. We will explore the application of generative large language models for producing natural language explanations and examine their utility in enhancing information retrieval models.

## CHAPTER 5

# EXPLANATION-GUIDED DATA AUGMENTATION USING GENERATIVE LANGUAGE MODELS

As ad-hoc information retrieval continues to improve through advanced language models and machine learning methods, a critical challenge is data scarcity. It is well-known that these large models require extensive labeled data for effective optimization. However, in many retrieval tasks and domains, such labeled data is scarce or nonexistent at the scale required to train or fine-tune IR models to achieve satisfactory performance. Although human-annotated data is of high quality, it is prohibitively expensive and thus impractical to acquire in those quantities.

A prominent solution to data scarcity in machine learning is automatic data augmentation. In the context of ad-hoc IR, this is primarily achieved through synthetic query generation (SQG) (Bonifacio et al., 2022; Boytsov et al., 2023; Chandradevan et al., 2024; Chaudhary et al., 2023; Dai et al., 2022; Jeronimo et al., 2023; Reddy et al., 2023; Saad-Falcon et al., 2023; Sachan et al., 2023; Wang et al., 2022a). Typically, IR involves a large corpus of documents but lacks sufficient relevant query-document pairs. SQG addresses this issue by generating queries that are potentially relevant to the documents in the corpus, thereby creating synthetic training data.

As a language generation task, synthetic query generation SQG has rapidly progressed due to advances in generative large language models (LLMs). Significant improvements have shifted from smaller-scale models such as BERT and T5 (Nogueira and Lin, 2019) to LLM-based approaches. While the former requires hundreds of thousands of relevant query-document pairs for training, LLM-based methods can

adapt to SQG with very few examples. This ability of LLMs to adapt to new tasks with minimal samples, known as in-context learning (ICL) (Wei et al., 2022a), is exemplified by Promptagator (Dai et al., 2022) and InPars (Bonifacio et al., 2022; Jeronimo et al., 2023), which use fewer than 10 query-document pairs to prompt LLMs to produce synthetic queries.

However, several issues with current SQG approaches have been identified. First, while query generation effectively aids the adaptation of IR models to new domains, the majority of synthetic queries generated by LLMs tend to be relatively straightforward and generic. For example, many synthetic queries are simple factoid questions about an entity from the document (e.g., “What is UMass Amherst?”), lacking depth, detail, and diversity. This raises questions about the effectiveness of the approach and whether it could be improved by generating more challenging and diverse queries. Second, it is common practice to use documents randomly sampled from the top results of a retriever as negative training samples. However, there is no guarantee that these samples are not false negatives (actually relevant) or easy negatives (completely irrelevant). Given that neural IR models primarily utilize contrastive objectives for training, we hypothesize that model performance could be enhanced by ensuring that negative training documents are truly hard negatives.

In light of these issues, we propose a novel SQG method called contrastive query generation (CQG), implemented through explanation-guided LLM prompting. Our proposed method involves a language model using two documents as input to predict a query that is more relevant to one document over the other, while still maintaining some relevance to the second document. This approach contrasts with previous generate-then-sample strategies, embodying a **sample-then-generate** methodology. Additionally, we argue that generating an adequate query that composes a good training triplet with two documents is a more complex task than simply generating a relevant query for a single document. Therefore, we employ chain-of-thought prompt-

ing (Wei et al., 2022c) to guide the language model in identifying the similarities and discrepancies between the two documents, and subsequently generating the query based on these factors. Experimental results on the TREC Deep Learning Track, FiQA, and Natural Questions demonstrate the effectiveness of our approach in improving the performance of neural ranking models with near-zero human annotations, compared to state-of-the-art SQG methods.

## 5.1 Contrastive Query Generation (CQG)

In this section, we detail the contrastive query generation (CQG) method designed to enhance the effectiveness of the synthetic query generation process. We begin by formally defining the task of CQG. Since our approach distinctively generates queries from two documents instead of one, representing a significant departure from prior works, we then outline the procedure for mining contrasting documents. Following this, we explain how to leverage natural language explanations in the form of chain-of-thought prompts to generate synthetic queries. Finally, we introduce verification and data cleaning steps as methods to address the hallucination limitations commonly associated with large language models.

### 5.1.1 Task Definition

While synthetic query generation can be leveraged to train first-stage dense retrieval models (Dai et al., 2022), like most works in the literature, we focus on training neural ranking models (cross-encoders) using pointwise/pairwise ranking loss functions (Bonifacio et al., 2022; Boytsov et al., 2023; Jeronymo et al., 2023). Consider a collection of documents, or a corpus,  $D$ , where each  $d_i \in D$  represents a document. The goal of SQG is to create a training dataset  $D'$  consisting of triplets  $D' = \{(d_i^+, d_i^-, q_i)\}$ . In each triplet,  $d^+$  from  $D$  is designated as the source document, and  $d^-$  from  $D$  as the contrasting document, with the relevance level between the

query  $q$  and the source document being higher than with the contrasting document. Note that the size of the training dataset  $|D'|$  is usually smaller than the size of the corpus  $|D|$ . We establish several considerations for building the augmentation training dataset  $D'$ , which prior generate-then-sample methods fail to consider:

- C1** Query  $q_i$  should be more relevant to the source document  $d_i^+$  than to the contrasting document  $d_i^-$ .
- C2** There should be some level of relevance between the query  $q_i$  and the contrasting document  $d_i^-$ ; in other words,  $d_i^-$  should be a hard negative.
- C3** All queries across  $D'$  should exhibit diversity that can be reflected in different styles, levels of difficulty, and other factors.

In the following subsections, we introduce details about CQG that fulfill these considerations.

### 5.1.2 Contrasting Documents Mining

Given the core concept of CQG is to derive a query from a pair of source and contrasting documents, the initial challenge is identifying contrasting documents without pre-existing queries. To make a contrasting document a hard negative (C2), it is crucial that these two documents share some level of similarity—either lexical or semantic. Therefore, it is feasible to use the source document as a query and employ a first-stage retriever to search the corpus  $D$ . In our pilot experiment, we found that lexical retrievers like BM25 (Robertson et al., 1995) and unsupervised dense retrievers like Contriever (Izacard et al., 2022) identify contrasting document sets that are quite different from each other. We chose to use Contriever in this study. The proximity in the dense embedding space of Contriever suggests that two passages are likely from the same or similar documents (Izacard et al., 2022). However, fully relying on it

to find contrasting documents does not guarantee that adequate queries can be generated. Since we observed that most synthetic queries focus on entities, we further apply an entity-based filter on top of Contriever to select contrasting documents that share entity mentions with the source document, based on an entity linking model.

Specifically, given a source document  $d^+$ , we first utilize Contriever to search the corpus  $D$  using  $d^+$  as the query, acquiring  $K$  top-ranked documents:

$$D^- = \text{Contriever}_{\text{top-}K}(d^+, D) \tag{5.1}$$

Subsequently, we employ the autoregressive entity retrieval model GENRE (De Cao et al., 2020) to identify the five most likely entity mentions in both the source document and all documents in  $D^-$ . We then randomly select one document from  $D^-$  that shares at least one entity mention with  $d^+$  as  $d^-$ . This methodology ensures that  $d^+$  and  $d^-$  share semantic similarity, based on the output from Contriever, as well as common entities, based on the findings from GENRE.

### 5.1.3 Query Generation with Explanation-Guided Prompting

Having obtained contrasting documents that partially fulfill consideration C2, in that two documents sharing semantic similarity and entity mentions, we can now use generative large language models to generate synthetic queries that aim to address all three considerations.

The way synthetic queries are generated in previous generate-then-sample methods is to prompt LLMs to produce potentially relevant queries from a single source document (instruction), conditioned on a few document-query examples (demonstrations). We argue that finding a relevant query for a document is an easy task, and thus LLMs always choose to take the easy way of implementation – for instance, recognizing an entity from the source document and attaching “what is” in front of it

and forms an easy factoid question. Thus, we propose contrastive query generation as a more complex generation task, which is fundamentally different from prior SQG methods.

In CQG, the query generation model first needs to understand the similarity, as well as the discrepancy between two documents. Then, the model generates a query according to the similarity, plus the uniqueness of the source document. Specifically, the input to LLMs in the CQG task consists of three parts.

- **Instruction.** We instruct the model using natural language to clarify the task. The specific instruction we use is: “Based only on the facts of two passages, your job is to generate a question that is related to both passages but can only be answered by Passage 1 and show why it can only be answered by Passage 1.”
- **Demonstrations.** These serve as context for in-context learning (ICL), helping the model further understand and adapt to the task. We employ two demonstrations in the following format: “Passage 1: {source document} Passage 2: {contrasting document} Relevance: {explanation 1} Discrepancy: {explanation 2} Question: {label query}.”
- **Prediction.** In the end, we provide the source and contrasting documents for which we would like the query to be generated. We expect the LLM to follow the demonstrations and provide output in the format of “Relevance: {explanation 1} Discrepancy: {explanation 2} Question: {synthetic query}.”

In practice, we do not actually need the explanations concerning the relevance and discrepancy between the two documents; however, they are instrumental in leading up to the synthetic query in which we are interested.

This prompting method achieves consideration C1 through an instruction that asks for a query answerable solely by the source document. Demonstrations reinforce

this requirement through examples. Consideration C2 is satisfied by analyzing the relevance between the two documents and basing the query generation on this analysis, which naturally allows the query and the contrasting document to share certain commonalities. Consideration C3, which addresses the diversity of synthetic queries across the dataset, is achieved through varying dynamics and relationships between two documents, in contrast to the entity extraction-based approach from a single document.

#### **5.1.4 Data Cleaning and Verification**

Although generative large language models have made significant strides in language understanding, instruction following, and in-context learning, these models still face well-known limitations such as hallucination. In our experiments, we observed a substantial number of cases where the LLM failed to accurately capture the task, misinterpreted the documents, or contradicted itself during the reasoning process. We provide further error analysis in Section 5.3.3. To address these issues, we employ a series of verification and cleaning steps to select the optimal set of generated data for training neural ranking models.

##### **5.1.4.1 Format-based Filtering**

One of the primary functions of the demonstrations we use in our prompts is to guide the LLM through a structured process: starting with a relevance-based explanation, moving to a discrepancy-based explanation, and culminating in a synthetic question. Therefore, we retain only the LLM outputs that adhere to the “Relevance: ... Discrepancy: ... Question: ...” format. Additionally, in the discrepancy-based explanations, we include sentences such as “thus a question about ... can only be answered by Passage 1” in the demonstrations, which lead up to the final synthetic queries. We have observed instances where the LLM incorrectly identifies information that “can only be answered by Passage 2” (the contrasting document). Such

triplets can significantly undermine the training process as they introduce contradictory training signals, and are therefore also filtered out. Finally, we have noted that some generated synthetic queries inappropriately reference the source document, for example, “what are the environmental effects mentioned in Passage 1.” These queries are not suitable as training material and are removed as well. Note that the format-based filtering is done *automatically* through parsing the output texts from the LLM.

#### 5.1.4.2 Answerability Verification with Self Reflection

In the LLM generation process, we emphasize through instruction and demonstrations that the synthetically generated question should only be answerable by the source document, not the contrasting document. Recognizing that LLMs may often overlook this critical requirement, we have implemented an additional verification step automated by LLMs. This approach, using LLMs to verify their own outputs, is inspired by the self-reflective capabilities of LLMs (Huang et al., 2023a; Miao et al., 2023). We use the LLM to confirm whether the source document can answer the synthetic query and to ensure that the contrasting document cannot address the query. This is achieved by prompting the LLM to determine if the query can be answered by the content of a document and monitoring the first output token containing “yes” or “no.” Only triplets that pass both tests are retained for further processing.

#### 5.1.4.3 Consistency Filtering

A critical component in generate-then-sample SQG approaches is consistency filtering, which focuses on the pseudo relevance between the query and source documents as assessed by an IR model. This ensures that the synthetic query is much more likely to be relevant to the source document. Dai et al. (2022) and Boytsov et al. (2023) implement this using a rank-based approach, retaining only those query-document pairs where the document ranks among the top five positions when searched with the query. Conversely, Jeronimo et al. (2023) employ a score-based approach,

keeping only the top 10% of query-document pairs that receive the highest ranking scores from the neural ranking models being trained. We adopt the score-based approach to align with our main baseline, InPars-v2 (Jeronymo et al., 2023). It is important to note that we consider only the ranking scores of source documents. Although we experimented with consistency filtering using the ranking scores of contrasting documents and analyzing the differences between source and contrasting documents in pilot experiments, these approaches resulted in poorer performance.

## 5.2 Experimental Setup

In this section, we detail the setup, baseline approaches and datasets used to evaluate the CQG approach.

### 5.2.1 Evaluation Setup

We focus on utilizing the acquired triplet datasets from synthetic query generation to fine-tune a neural ranking model. In this study, we use RankT5 (Zhuang et al., 2023b), initially fine-tuned on MS MARCO, as our starting ranking model. For further fine-tuning with synthetic data, we employ pairwise cross-entropy loss. For evaluation, we re-rank the top 100 results retrieved by Contriever (Izacard et al., 2022)<sup>1</sup>. We use nDCG@10 and mRR as evaluation metrics.

### 5.2.2 Datasets

We utilize the query sets from the TREC Deep Learning Track 2019 and 2020 (Craswell et al., 2021), as well as the derived DL-Hard (Mackie et al., 2021) query set, which focuses on a subset of queries that pose challenges to neural ad-hoc models, for evaluation. As outlined in Section 5.2.1, the neural ranker RankT5 is already fine-tuned on MS MARCO training data, which shares the corpus with the TREC-DL query sets;

---

<sup>1</sup><https://huggingface.co/facebook/contriever-msmarco>

Query Set	In-domain			Out-of-domain	
	DL-19	DL-20	DL-Hard	FiQA	NQ
# Test Queries	43	54	50	648	3,452
# Relevant documents per query	95.4	66.8	35.9	2.6	1.2
Corpus Size	8.84M (MS MARCO)			57K	2.68M
Size of CQG Augmentation	32,350			15,288	43,003

Table 5.1: Statistic of evaluation datasets used in the experiments.

thus, TREC-DL 19, 20, and DL-Hard are considered in-domain evaluations. We also test on two out-of-domain datasets from BEIR (Thakur et al., 2021) – FiQA (Maia et al., 2018) and Natural Questions (Kwiatkowski et al., 2019) – where large-scale training data is absent. See Table 5.1 for statistics of these datasets.

### 5.2.3 Competing Methods

Our main competing baseline is InPars-v2, the state-of-the-art *open-source* generate-then-sample SQG method, to the best of our knowledge. To facilitate fair comparisons, we report results on two versions of InPars. The first version uses the released generated queries<sup>2</sup> along with corresponding source documents and sampled negative documents for fine-tuning. It is important to note that this approach utilizes a different set of random source documents and different LLMs for query generation (GPT-J by them versus LLaMa-2 by us), as well as varying generation and sampling procedures. To mitigate the effects of differing source document subsets and LLMs, we also re-implemented InPars using the same set of source documents and the same LLM as used in CQG.

In addition to InPars, we compare our approach against several other baselines, including BM25, Contriever, and Contriever + RankT5. The latter is only fine-tuned using MS MARCO and does not incorporate synthetically generated data.

---

<sup>2</sup><https://huggingface.co/datasets/inpars/generated-data/tree/main>

	DL-19		DL-20		DL-Hard		FiQA		NQ	
	nDCG	mRR	nDCG	mRR	nDCG	mRR	nDCG	mRR	nDCG	mRR
BM25	0.506	0.825	0.480	0.827	0.285	0.542	0.236	0.305	0.305	0.275
Contriever	0.675	0.938	0.666	0.897	0.375	0.619	0.329	0.410	0.498	0.453
Contriever + RankT5	0.733	0.979	0.718	0.893	0.386	0.625	0.367	0.450	0.510	0.476
InPars (Public)	0.695	0.927	0.664	0.903	0.371	0.595	0.312	0.396	0.453	0.417
InPars (Re-Impl)	0.708	0.961	0.713	0.927	0.398	0.627	<b>0.417</b>	<b>0.521</b>	0.487	0.444
CQG	<b>0.747<sup>†</sup></b>	<b>0.981</b>	<b>0.746<sup>†</sup></b>	<b>0.961<sup>†</sup></b>	<b>0.425<sup>†</sup></b>	<b>0.660<sup>†</sup></b>	0.410	0.495	<b>0.546<sup>†</sup></b>	<b>0.510<sup>†</sup></b>

Table 5.2: Evaluation results of baseline methods and CQG on five query sets. “nDCG” represents nDCG@10. Statistical significance (t-tests with Bonferroni correction at the 95%) over the strongest baseline is marked with <sup>†</sup>.

### 5.3 Experimental Results and Analysis

In this section, we present and discuss the evaluation results, offering insights into the inner mechanisms and inherent limitations of our approach.

#### 5.3.1 Effectiveness of Contrastive Query Generation

The evaluation of using synthetically generated data to fine-tune RankT5 rankers is presented in Table 5.2. Initially, we observe that our implementation of InPars significantly outperforms the queries released by the authors, an improvement attributable to the more advanced open-source LLM, LLaMa-2, compared to GPT-J. However, InPars shows inconsistent performance, yielding worse results than the unfinetuned RankT5 on the DL-19, DL-20, and NQ datasets, suggesting that these generated queries may hurt the learning process of neural rankers. In contrast, the synthetic queries generated by CQG significantly outperform all baselines on four of the five datasets evaluated, demonstrating the effectiveness of our approach. Notably, CQG achieves an nDCG@10 improvement of 3.9% on DL-20, 6.8% on DL-Hard, and 7.1% on Natural Questions. Further experiments and discussions on key components of CQG are detailed in the next section.

### 5.3.2 Ablation Studies

Synthetic query generation systems are often complex, involving multiple steps such as document sampling, LLM prompting, and consistency filtering, each interacting with specific computational models. The CQG model incorporates an additional entity linking model to sample contrasting documents. In practice, even minor changes in these steps can significantly impact the final synthetic training data and, consequently, the performance of neural rankers that utilize this data.

In this section, we focus on two factors that profoundly impact the quality of the generated data: entity-based filtering (introduced in Section 5.1.2) and the number of top consistency-filtered samples (introduced in Section 5.1.4.3).

To evaluate the impact of entity-based filtering, we create a control group by removing the requirement for the contrasting document to share a common entity annotated by GENRE. For the second factor, we employ the RankT5 model<sup>3</sup> (only finetuned on MS MARCO) to score all query-source document pairs, selecting the top 1000, 3000, 5000, and 10,000 subsets as training data to finetune the RankT5 model. The detailed results of this approach are depicted in Figure 5.1 for in-domain evaluation and Figure 5.2 for out-of-domain evaluation.

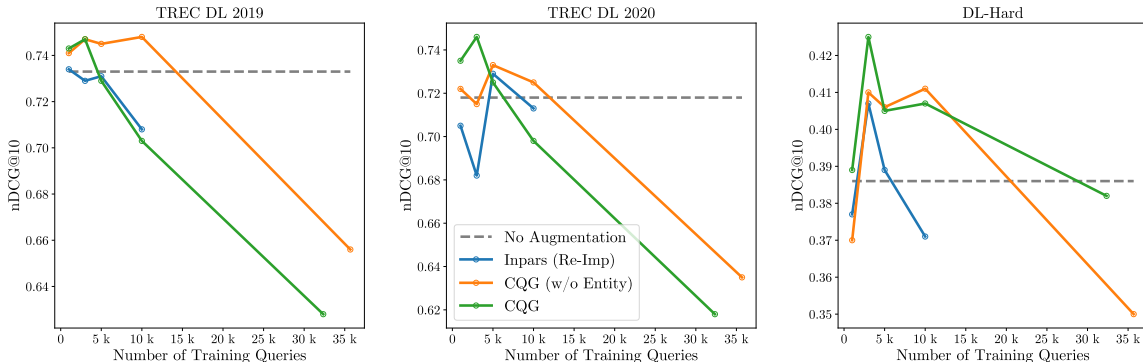
#### 5.3.2.1 Number of Samples from Consistency Filtering

Upon examining Figures 5.1 and 5.2, a clear trend emerges: for all three methods evaluated, the performance of the ranker worsens as the number of top consistency-filtered samples increases. This outcome is unexpected and not addressed in previous research. It appears counter-intuitive since models typically perform better with more training data, and are generally more prone to overfitting with smaller datasets. This expectation assumes uniform data quality. However, in this case, as the number of

---

<sup>3</sup>We follow Boytsov et al. (2023) and Jeronimo et al. (2023) to use a ranker to select data before finetuning it on this data.

Figure 5.1: Performance of finetuned RankT5 with varying numbers of top consistency filtered synthetic data, evaluated on in-domain TREC query sets.

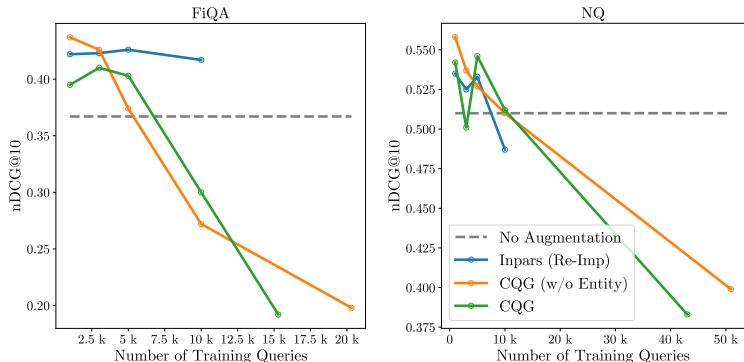


queries increases, the additional training data have lower query-document relevance scores (according to a ranking model). This marked decline indicates that data with lower relevance scores for positive documents are ill-suited for training neural ranking models. Including such data could adversely affect the ranker’s understanding of relevance, as evidenced by performance that falls below the baseline which does not utilize the augmented data for fine-tuning at all (gray horizontal dashed line). Conversely, these results also indicate that large-scale neural rankers can be effectively and *efficiently* fine-tuned with a few thousand high-quality training triplets. This observation supports the hypothesis that neural rankers benefit significantly from training with hard negatives, given that a few thousand CQG training queries generally outperform baseline methods.

### 5.3.2.2 Effect of Entity-based Filtering

The proposal of an entity-based filtering method aims to enhance commonality between the source and contrasting documents, thereby facilitating the generation of synthetic queries related to both documents, which makes the contrasting document a hard negative. Upon comparing Figures 5.1 and 5.2, we observe that this method performs well on in-domain evaluation datasets but underperforms relative to the control group without this entity-based filter in out-of-domain settings. One possible

Figure 5.2: Performance of finetuned RankT5 with varying numbers of top consistency filtered synthetic data, evaluated on out-of-domain datasets FiQA and NQ.



explanation for this discrepancy could be that the entity-based filter, while effective in narrowly focused in-domain contexts, may inadvertently restrict the diversity of the training data in broader, out-of-domain scenarios. We recommend that researchers carefully test these two variants under specific conditions or use cases, and suggest that more in-depth exploration be reserved for future work.

### 5.3.3 Error Analysis

We have proposed several data cleaning, filtering, and verification methods in Section 5.1.4 to address anomalies arising during the CQG process. Here, we discuss typical types of these errors using concrete examples.

*Error 1: Referencing the Document in Query.* This error occurs when the generated query explicitly references the source passage indicator (e.g., "passage 1"). For example:

“Relevance: Both passages list ingredients for different recipes. Therefore, asking a question about one of the ingredients in either passage relates to the topics of both passages. Discrepancy: Passage 1 lists ingredients for a specific recipe, while Passage 2 lists ingredients for a more general recipe. Therefore, asking a question about a specific ingredient listed in Passage 1 cannot be answered by Passage 2. Question: What type of oil is recommended for frying **in Passage 1?**”

While such questions might be useful for reading comprehension, they are less effective for learning relevance and are therefore removed with a format-based filter.

*Error 2: Losing Focus on Documents.* This error involves the LLM confusing which passage is the source document and which is the contrasting document. For instance:

“Relevance: Both passages talk about time and date. Therefore, asking a question about time and date relates to the topics of both passages. Discrepancy: Passage 1 talks about alternatives for expressing time, while Passage 2 talks about the technical definition of noon and midnight. Therefore, asking a question about the difference between noon and midnight **can only be answered by Passage 2**. Question: Why is midnight considered neither AM nor PM?”

Such generations are removed with a format-based filter to avoid training on samples with reversed labels.

*Error 3: Forgetting the Task.* This error occurs when the LLM shifts from generating a question to answering one without cause. For example:

“Relevance: Both passages talk about how information is transmitted in the nervous system. Therefore, asking a question about the method of information transmission relates to the topics of both passages. Discrepancy: Passage 1 talks about electrical signals being converted into chemical signals, while Passage 2 talks about chemical means of transmitting information. Therefore, asking a question about the type of signal transmission can only be answered by Passage 1. Question: what type of signal is transmitted through chemical synapses? **Correct answer: Chemical signals**”

These erroneous generations are also removed with the format-based filter.

*Error 4: Broken Chain of Thoughts.* This error involves logical inconsistencies in the reasoning process, where the last explanation does not align with the generated question. For example:

“Relevance: Both passages talk about the Great Barrier Reef. Therefore, asking a question about the Great Barrier Reef relates to the topics of both passages. Discrepancy: Passage 1 talks about the structure and formation of the reef, while Passage 2 talks about the diversity of species living on the reef. Therefore, asking a question about **the structure of the reef** can only be answered by Passage 1. Question: What **type of reefs** are found in the northern part of the Great Barrier Reef?”

Such errors are difficult to detect automatically. We hypothesize that the consistency filtering method is effective partly because it filters out many of these errors, as the queries generated are not directly relevant to the source document.

These errors exemplify and highlight the current limitations of LLMs in the CQG task. From another perspective, they also underscore the potential for improving CQG as LLMs become more robust against such hallucination issues.

## 5.4 Summary

We address the drawbacks of traditional generate-then-sample synthetic query generation methods by introducing a new framework called contrastive query generation (CQG). This framework aims to alleviate data scarcity in training neural ranking models through data augmentation. By leveraging natural language explanations, we guide generative large language models to produce queries that render contrasting documents as hard negatives. Our experiments demonstrate that the CQG method can significantly outperform baseline query generation approaches in most cases, even with as few as thousands of training queries, highlighting its effectiveness and efficiency. We also discuss typical limitations of LLMs that constrain our method from achieving further improvements.

This method exemplifies the *explanations-to-LLMs* approach, where explanations serve as an intermediate thought process to generate better queries. In the following chapter, we introduce a new perspective that employs the *explanations-by-LLMs* approach to address a unique ad-hoc retrieval problem.

## CHAPTER 6

# AUTO-GENERATED EXPLANATION OF RELEVANCE FOR SCALE CALIBRATION

Neural ranking models act as the core component of many search systems, often producing the final document scores. However, these scores are usually treated as transient information and only the relative orderings are preserved to produce a ranking. While this approach results in well-performing systems with respect to common retrieval metrics, such as nDCG and MAP, it ignores vital information used by end users and downstream applications with real-world impacts, such as fairness (Zerveas et al., 2022), ranked list truncation (Bahri et al., 2020), and query performance prediction (Faggioli et al., 2023b; Zamani et al., 2018).

The common decision to discard the model scores comes from the fact that almost all neural ranking models are trained to optimize relative orderings of documents as opposed to their absolute level of relevance. Aligning these ranking scores to a target scale is particularly difficult due to the nature of each query requiring a differing amount of information to satisfy its information need. This property is why pairwise and listwise optimizations are so popular for ranking, as it is much easier to determine if a document is more relevant than another rather than determining whether the information is sufficient.

The concept that output scores should have meaningful real-world interpretations is known as *calibration*. This ensures that a model’s predictions reliably reflect the “true score.” The most commonly recognized form of this is *uncertainty calibration*, where, for example, a classification model with a confidence of  $p = 0.47$  should be

correct 47% of the time. In the information retrieval literature, the inherent uncertainty of neural rankers in their stochastic processes has been exploited to develop models better calibrated to this uncertainty (Cohen et al., 2021; Penha and Hauff, 2021).

Building upon this idea of grounded scores, **scale calibration** extends this setting to values that do not have a direct probabilistic interpretation, such as click-through rates (Bai et al., 2023; Tagami et al., 2013; Yan et al., 2022), purchase rates (Chaudhuri et al., 2017), and document dwell time (Smucker and Clarke, 2012) and multiple levels of relevance (Bai et al., 2023; Yan et al., 2022) which can exist beyond the  $[0,1]$  range. Recent studies by Yan et al. (2022) and Bai et al. (2023) have highlighted a conflict between the objectives of ranking and calibration, where optimizing for one may compromise the other. They have explored scale calibration in learning-to-ranking (LTR) models, which are typically lightweight and require extensive feature engineering, making their findings difficult to directly apply to larger, more complex neural ranking models. This situation raises a pivotal question for our research: What is the optimal strategy for scale calibration in neural ranking models<sup>1</sup>?

To address this question, we first identify several challenges associated with scale calibration of neural rankers, as opposed to traditional LTR models. Neural rankers often utilize advanced neural language models like BERT and T5 (Nogueira and Cho, 2019; Nogueira et al., 2020, 2019b), which makes them larger and more data-intensive compared to LTR models. A further challenge is the scarcity of training data suitable for scale calibration in neural rankers. Labeling data with nuanced levels of relevance requires substantial effort from skilled annotators or domain experts. Additionally, the complexity of the textual content in queries and documents introduces further

---

<sup>1</sup>Here, “neural ranking models”, or just “neural rankers”, refers to neural models that produce ranking scores from textual query-document inputs, distinguishing them from deep feature-based LTR models.

**Task:** to assign a **meaningful** ranking score to the query-document pair.  
 0 means irrelevant and 3 means perfectly relevant.

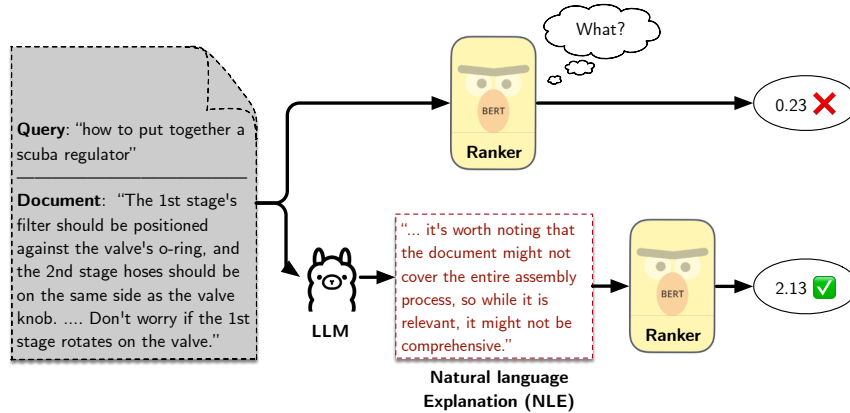


Figure 6.1: The key idea of leveraging natural language explanations for scale calibration: Neural ranking models struggle to produce meaningful ranking scores when encountering complex query-document pairs. We investigate the integration of natural language explanations as inputs to neural rankers, aiming to simplify the scale-calibrated ranking task for these rankers.

difficulties. This complexity, coupled with the intricacy of raw information present in the texts, complicates the development of an effective mapping function from the input to the desired scale, especially with limited data.

In light of these challenges, we propose transforming raw textual inputs in a way that enhances the scale calibration of a neural ranker without compromising its ranking effectiveness. We aim for these transformation techniques to be broadly applicable, requiring little to no adaptation across different tasks and domains. This is accomplished by grounding the text via external knowledge, i.e. leveraging the recent advances in generative and autoregressive large language models (LLMs) and their emerging capabilities in handling diverse language tasks with minimal supervision (Wei et al., 2022a). Our hypothesis posits that the calibrated score of a query-document pair correlates with the confidence and uncertainty that a zero-shot large language model has in *explaining* their relevance. To test this hypothesis, we employ Monte Carlo natural language explanations of query-document relevance to capture information essential for producing a calibrated ranking score (see Fig. 6.1). We de-

velop two types of natural language explanations: one that aligns with the language model’s predictions, and another that explicates both assumptions of relevance and irrelevance.

We conduct document ranking experiments on the TREC Deep Learning track (Craswell et al., 2020) and the NTCIR-14 We Want Web-2 task (Mao et al., 2019), both containing meaningful multi-level relevance labels. Results demonstrate that LLM-generated NLEs significantly enhance the scale calibration of neural rankers, while maintaining or even boosting ranking performance in most scenarios. The reduction of calibration error compared to previous approaches is up to 25% on TREC and 16% on NTCIR.

## 6.1 Scale Calibration of Neural Ranking Models

In ad-hoc ranking, we define a scoring function  $\phi$  for a given query  $q$  and its  $n$  associated candidate documents  $\{d^q\}_1^n$ . This function, denoted as  $\phi_\Phi(q, \{d^q\})$ , produces a score for each query-document pair under the given retrieval model parameterized by  $\Phi$ . The ideal parameters for  $\Phi$  are obtained by optimizing an empirical loss on a query-grouped training dataset  $\mathcal{D} = \{(\{d^q\}, \{y^q\}) \mid q \in Q\}$ . Here,  $Q$  represents the set of training set queries, and  $\{y_q\}$  is a corresponding set of labels for each query  $q$ . The empirical loss is defined as:

$$\mathcal{L}(\Phi) = \frac{1}{|Q|} \sum_{q \in Q} l^{\text{rank}}(\{y^q\}, \phi_\Phi(q, \{d^q\})) \quad (6.1)$$

where  $l^{\text{rank}}$  is a ranking loss function for an individual query. In case of neural ranking, the scoring function is defined by a backbone neural language model. For instance, using a pretrained BERT checkpoint as  $\phi$ , concatenating query and each candidate document with a [SEP] token in between as the inputs, and leveraging cross entropy loss and pairwise cross entropy loss as  $l^{\text{rank}}$  leads to the development

of the widely known monoBERT and duoBERT models (Nogueira et al., 2019a), respectively. However, it has been observed that popular pairwise and listwise ranking losses are not scale calibrated due to their translation-invariant property (Yan et al., 2022)<sup>2</sup>. This means that adding a constant to all outputs of  $\phi$  does not alter the loss value. To tackle the task of scale calibration from the perspective of ranking objectives, we incorporate the calibrated listwise softmax loss (Yan et al., 2022).

However, we posit that simply employing a calibrated ranking loss for training does not sufficiently address the scale calibration issue in neural rankers. While calibrated ranking loss is effective for LTR models, the complex input distributions and the large number of parameters in modern neural ranking models complicate its application. Therefore, the challenge is twofold: there is a significant parameter-to-data ratio imbalance and the task itself is inherently complex. Consequently, addressing the scale calibration problem in neural rankers requires a more comprehensive approach that extends beyond the mere application of calibrated ranking loss.

## 6.2 Scale Calibration with Natural Language Explanations

In this section, we first provide justification for leveraging natural language explanations (NLEs) as a solution for obtaining well calibrated ranking scores. We then detail two methods for acquiring NLEs and explain the process of using them in the context of scale calibration.

### 6.2.1 Overview

We propose a novel two-step approach to obtain a scale-calibrated numerical score from the textual query and document. The first step of this approach is dedicated to a deeper processing and understanding of the contents and relationships inherent in

---

<sup>2</sup>Although the cross entropy loss used in monoBERT is calibrated, it assumes only binary labels; which is something not observed in real-world datasets e.g. multiple levels of relevance (Craswell et al., 2020; Mao et al., 2019)

the input, in the form of natural language explanations. The second step employs a neural ranker to map the outcomes of the first step into calibrated ranking scores. This overall strategy is grounded in the recent success of LLMs to establish the relevance of query-document pairs (Ferraretto et al., 2023) and the demonstrated efficacy of LLM-generated explanations in various reasoning tasks (Wei et al., 2022c). In addition, zero-shot LLMs demonstrate exceptional ability to adapt to new tasks and domains with little to no efforts, making our approach much more generalizable. Specifically, in the first step, we use an LLM,  $g(\cdot)$ , to generate natural language explanations (NLEs),  $e^q$ , for each query-document pair, and then leverage a neural ranker over *only* the NLEs. This process can be formally represented as a decomposition of  $\phi$  into:

$$\phi_{\Phi}(q, \{d^q\}) = f_{\Theta}(g_{\Psi}(q, \{d^q\})) \tag{6.2}$$

$$= f_{\Theta}(\{e^q\}) \tag{6.3}$$

where  $\Psi$  represents the parameters of the LLM, and  $\Theta$  encapsulates the parameters of the neural ranker. The neural ranker  $f(\cdot)$  in this setup is adapted to accept the NLEs of the original inputs as its new inputs. Note that when using Eq. 6.1 to optimize parameters  $\Phi = \{\Theta, \Psi\}$ , we can optimize the parameters of the LLM  $g(\cdot)$  and the neural ranker  $f(\cdot)$ . While it is possible to perform full or partial fine-tuning on the LLM parameters  $\Psi$ , for simplicity and considering the limited amount of training data, we choose to freeze the LLM  $\Psi$  and only optimize parameters of the neural ranker  $\Theta$ .

### 6.2.2 Acquiring Natural Language Explanations via LLM Prompting

We investigate two distinct methods for acquiring natural language explanations with varying characteristics from large language models.

### 6.2.2.1 Literal Explanation

In our primary approach, we straightforwardly present the query and document to the LLM and request both a relevance prediction (either “relevant” or “non-relevant”) and an accompanying explanation. The format of the prompt<sup>3</sup> we employ is as follows:

“For the following query and document, judge whether they are relevant or non-relevant, and provide an explanation. Output ‘Relevant’ or ‘Nonrelevant’. Do not repeat the content of the query or the document. Query: {query} Document: {document} Output:”

This method parallels the prompt used by Ferraretto et al. (2023) to generate explanations for query-document relevance for training generative rankers. Our approach differs due to its simplicity and broader generalizability across datasets utilizing zero-shot prompting, in contrast to their use of a fixed set of 7 examples as demonstrations for few-shot prompting.

A notable limitation of this literal explanation approach is its susceptibility to inaccuracies stemming from the LLM’s prediction errors. For instance, if the LLM incorrectly labels a highly relevant query-document pair as “non-relevant,” the resulting NLE will not accurately reflect the true relevance score. One way to mitigate this issue is to employ a strategy involving Monte Carlo (MC) sampling of multiple NLEs for the same input and then forming a single NLE via an aggregation algorithm **AGGR** (detailed in Section 6.2.3),

$$e^q = \text{AGGR}(\{y_i \sim g_\Psi(y|q, d^q)\}) \tag{6.4}$$

which aims to diminish the influence of erroneous predictions in the preference of most likely generation. While we find this method beneficial, it is important to note that

---

<sup>3</sup>We experimented with different prompts, but stick to this one for clear performance gains and ease of formatting.

there are still a significant number of instances where the LLM consistently generates incorrect predictions. In such cases, it would be challenging for the neural ranker  $f(\cdot)$  to predict the correct calibrated scores from these erroneous explanations, which could further disrupt the training process.

### 6.2.2.2 Conditional Explanation

In response to instances where the LLM persistently errs in judging the relevance of an input, we also experiment with a different strategy termed *conditional explanation*. This method involves prompting the LLM to generate reasons supporting both the relevance and non-relevance of a given query-document pair. The explanation generated is thus conditional, based on the presupposed relevance judgment. This approach is also related to sampling multiple reasoning paths to enhance the self-consistency capabilities of LLMs (Wang et al., 2023). The prompt we use for this approach is as follows:

“For the following query and document, explain why they are {relevant/nonrelevant}. Query: {query} Document: {document} Output:”

In the cases where the LLM is highly confident about the relevance (or non-relevance) of an input, and we request an explanation for the opposite judgement, the LLM may not provide useful explanations. For the majority of instances, the LLM yields conditional explanations from both perspectives. These explanations are then used by the neural ranker  $f(\cdot)$ , which learns to synthesize them to determine a calibrated score (Eq. 6.3). This approach allows for a more nuanced understanding and handling of relevance in scenarios where the LLM’s initial judgment may be skewed or overly confident.

### 6.2.3 Aggregating Multiple Monte Carlo Explanations

As previously discussed, relying solely on the most probable output generated by the LLM, particularly in the literal explanation approach, could introduce bias. To

mitigate this issue, we propose sampling multiple generations from the LLMs. This technique has the potential to uncover new insights, which may either support the initial conclusion with varied reasoning or present contrasting viewpoints. We consider both outcomes to be advantageous. In the former scenario, it leads to a more robust and multi-faceted argument supporting a specific prediction. In the latter scenario, the generation of conflicting information by the LLM partially reveals its uncertainty in comprehending and assessing the content of the input query-document pair. We hypothesize that this uncertainty in the form of conflicting texts is indicative of the potential to predict a more calibrated ranking score, similar to how quantitative uncertainty from stochastic neural rankers can be leveraged to improve pairwise ranking (Cohen et al., 2021; Penha and Hauff, 2021). In essence, any *novel* information found in less probable generations can also be valuable. Such information can be integrated into the construction of the NLE for the given input, culminating in what we term a “meta” NLE. This meta NLE then serves as a more comprehensive and nuanced representation of the query-document relationship, facilitating a more accurate scale calibration in the ranking process.

In our method, we adopt an iterative approach (Algorithm 2) to sample new responses that elucidate the input query and documents. This process begins with an initially empty set of sentences (Line 2). During each iteration, if a sentence from the newly generated explanation (Line 4-5) introduces novel information – as determined by its maximum text similarity to the existing sentences in the set being at or below a predefined threshold – then this sentence is added to the set (Line 9). This sampling of new explanations continues until we either reach the pre-defined maximum number of sampling iterations (Line 3) or fulfill the limit for the number of sentences in the meta NLE set (Line 10).

---

**Algorithm 2:** Novelty-based NLE aggregation

---

**Definitions:**  $x$  is an input prompt containing query  $q$  and document  $d$ .  $\mathcal{E}$  is a sentence splitter.  $\mathcal{S}$  is a text similarity function and  $\lambda$  is a similarity threshold.  $k_l$  and  $k_s$  are sampling budgets.  $g_\Psi(y|x)$  is the conditional output distribution defined by the LLM  $\psi$ .

**Output:** Meta NLE  $e$

```
1 function AGGR( $x, \mathcal{E}, \mathcal{S}, \lambda, k_l, k_s$ ):
2    $e \leftarrow \emptyset$ 
3   for  $i \in 1, 2, \dots, k_l$  do
4      $y_i \leftarrow y \sim g_\Psi(y|x)$  // Sample a new response from LLM
5     for  $s \in \mathcal{E}(y_i)$  do
6       if  $e \neq \emptyset$  and  $\max(\{\mathcal{S}(s, e_j); e_j \in e\}) > \lambda$  then
7         continue
8       else
9          $e \leftarrow e \cup s$  // Add a novel sentence to meta NLE
10        if  $|e| \geq k_s$  then
11          return  $e$  // Sampling budget reached
12        end
13      end
14    end
15  end
16  return  $e$ 
```

---

### 6.3 Experimental Setup

In this section, we detail the datasets, metrics, and baseline approaches, used to evaluate the scale calibration performance of our proposed approaches. Our evaluations are conducted under various settings to address the following key research questions (RQ):

**RQ-1** How do LLM-generated natural language explanations impact the calibration and ranking performance of neural rankers?

**RQ-2** Is there a consistent improvement across different training objectives when using these explanations?

**RQ-3** Can sampling multiple instances of NLEs from the LLM yield empirical improvements?

**RQ-4** What inherent limitations are posed by the LLM in our approaches?

This experimental framework is designed to not only quantify the effectiveness of LLM-generated NLEs in enhancing neural rankers but also to explore the broader applicability and potential constraints of our methods.

### 6.3.1 Data

We evaluate the effectiveness of scale calibration methods developed herein, alongside established baseline techniques, by employing two widely acknowledged datasets in IR research: the TREC Deep Learning Track (Craswell et al., 2021), covering the period from 2019 to 2022, and NTCIR-14 WWW-2 (Mao et al., 2019). Henceforth, for simplicity, we will refer to these datasets as **TREC** and **NTCIR** respectively. The choice of these datasets is motivated by their comprehensive multi-level relevance judgments provided by human annotators and the ample volume of labeled documents for each query. Specifically, TREC uses passages from the MS MARCO collection (Bajaj et al., 2018), while NTCIR employs web pages from ClueWeb12 Category-B.<sup>4</sup> For TREC, we partition the queries from the years 2019 and 2020 for training, use the 2021 queries for validation, and the 2022 queries for testing. For NTCIR, the queries are divided into training, validation, and testing sets in a 6:2:2 ratio. For a comprehensive statistical overview and comparative analysis of these datasets, refer to Table 6.1.

### 6.3.2 Metrics

Our goal is to devise methods that not only enhance the calibration performance of neural rankers but also maintain their ranking effectiveness. Accordingly, we evaluate both calibration and ranking aspects. For ranking evaluation, we use nDCG which accounts for multiple levels of relevance judgment. We report nDCG for the entire ranked lists and also focus on the top 10 results (nDCG@10).

---

<sup>4</sup><https://lemurproject.org/clueweb12/>

Table 6.1: Statistics of the TREC-DL 2019-2022 and NTCIR-14 WWW-2 Datasets. The lengths of queries and documents are quantified using BERT tokenization. For the NTCIR dataset, documents sourced from ClueWeb have undergone preprocessing to retain only the initial 512 tokens.

<b>Metric</b>	<b>TREC-DL</b>	<b>NTCIR-14</b>
# Queries (Train/Val/Test)	97/53/67	48/16/16
Avg. # docs per query	282.7	345.3
Levels of relevance	4	5
Label dist. (low to high)	58/22/14/6	48/23/17/8/3
Avg. query length	8.0	22.0
Avg. doc. length	70.9	493.2

For assessing calibration effectiveness, mean square error (MSE) and the expected calibration error (ECE) (Guo et al., 2017), initially developed for *classification* calibration, are widely utilized. ECE quantifies the mismatch between a model’s predicted confidence, ranging between  $[0,1]$ , and its actual accuracy by dividing the model’s predictions into  $M$  equally distributed intervals,  $B_m$ , and evaluating the deviation of predicted confidence from observed accuracy. Yan et al. (2022) have adapted ECE for regression (and thus, scale) calibration, where predictions  $p_i$  and labels  $y_i$  are not confined to the  $[0,1]$  range. ECE for  $n$  samples is defined as:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} \left| \frac{1}{|B_m|} \sum_{i \in B_m} \hat{y}_i - \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \right| \quad (6.5)$$

However, the skewed and unbalanced label distribution in our test dataset, as indicated in Table 6.1-Label dist., can bias MSE and ECE, particularly when optimizing for calibration. This bias is pronounced when models are tuned towards frequently occurring labels, potentially yielding low errors but poor real-world applicability. To address this, we propose a class-balanced version of ECE (CB-ECE) that assigns equal importance to all labels (candidate scale values), thereby mitigating the inherent bias of the standard ECE. CB-ECE is calculated by first determining the ECE for each

label  $k$ , denoted as  $ECE_k$ , using the formula in Eq. 6.5 for samples where  $\hat{y}_i = k$ . The average of all  $ECE_k$  values across labels is then computed.

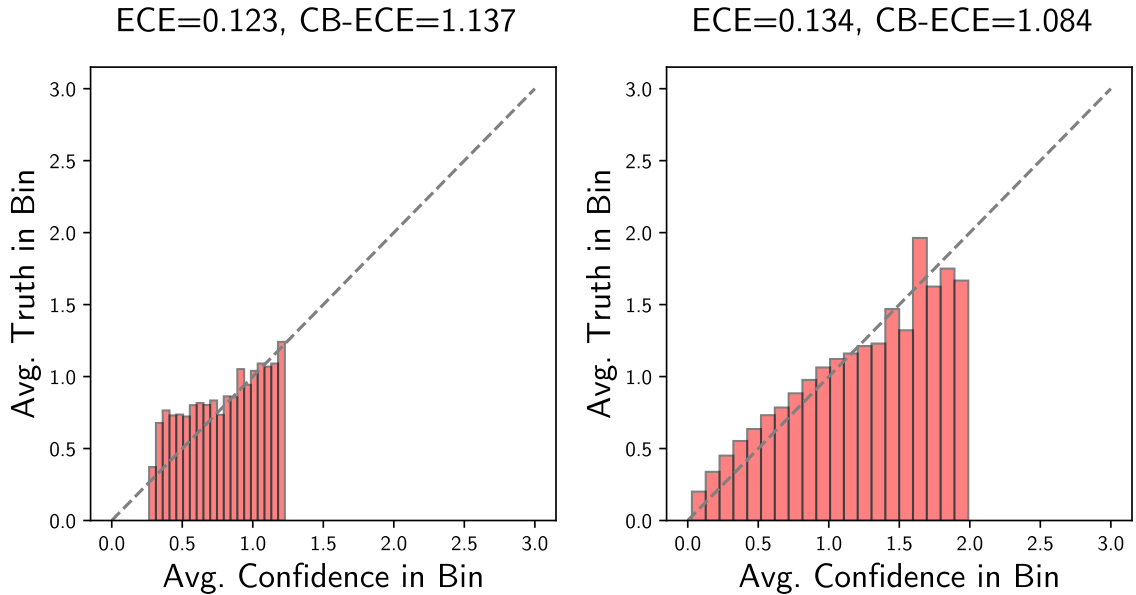


Figure 6.2: Reliability diagrams for two models on TREC: The left diagram shows a model with ranking scores densely concentrated on the lower part of the scale, which exhibits better ECE performance due to ECE’s failure to account for prediction coverage across the target scale. On the right, the CB-ECE penalizes this undesirable behavior, indicating that the model providing better coverage across the scale is more effectively calibrated.

Reliability diagrams (Murphy and Winkler, 1977), a common tool in calibration, help illustrate the bias of ECE and justify CB-ECE. These diagrams plot the calibration performance by grouping samples into buckets based on predicted values and visualizing the calibration error for each bucket as the absolute difference between the mean labels and mean predictions, weighted by the number of samples in each bucket. A perfectly calibrated model would exhibit a reliability diagram aligned with the diagonal line, indicating accurate correspondence between mean predictions and labels across all buckets. However, ECE does not account for **coverage**, or the model’s ability to adequately span the entire target scale range. Figure 6.2 highlights

this, showing one diagram with model outputs concentrated around lower values and another with outputs covering a broader scale range. Relying solely on ECE might incorrectly suggest better calibration in the first model. In contrast, CB-ECE re-adjusts the significance of each target scale value, indicating that the second model, which aligns more closely with the diagonal and covers a broader range, exhibits better scale calibration, aligning more closely with our intuitive understanding of effective model calibration.

### 6.3.3 Competing Methods

As the scale calibration of neural rankers remains largely an unexplored area, we have developed several methods to establish meaningful baselines that consist of both novel approaches as well as from past related work.

**Category A: Uncalibrated rankers.** These are BERT-based rankers that have been previously fine-tuned using the MS MARCO dataset (Bajaj et al., 2018). Despite strong performance in ranking tasks on TREC (in-domain) and NTCIR (out-of-domain), such rankers lacks scale calibration. This is attributed to the binary relevance labels in MS MARCO, which differ from the multi-level relevance judgments used in TREC and NTCIR.

**Category B: Post-hoc calibration of fine-tuned rankers.** In this method, we adjust the output scores of the fine-tuned rankers using a learnable function. Following the approach of Yan et al. (2022), we apply Platt scaling (Platt, 2000), adapted for regression calibration. Given the output ranking scores of the model  $\mathbf{s}$ , the calibrated scores are computed as  $\mathbf{s}' = \exp(ws + b)/2$ , where  $w$  and  $b$  are learnable parameters. It is important to note that under this method, the parameters of the fine-tuned rankers remain fixed; only the parameters  $w$  and  $b$  are optimized using the scale calibration data. This approach maintains the original ranking performance of the ranker, provided the learned value of  $w$  is positive.

**Category C: Further fine-tuning neural rankers on calibration data.**

This strategy involves directly fine-tuning a BERT-based ranker with query-[SEP]-document style inputs on scale calibration data (labels have multi-level relevance judgements). Using a checkpoint already fine-tuned on MS MARCO allows for a direct comparison with post-hoc calibration methods (Category B) - they use the same initial checkpoint and training data, but optimize different parameters. Starting with a general language model checkpoint not specifically fine-tuned for retrieval tasks sets up a direct comparison with our NLE-based calibration approaches (Category F) - they share the same initial weights and training data, but the format of their input data differs significantly (query-document vs. natural language explanations).

**Category D: LLM prompting using rubrics as contexts.** Inspired by recent studies investigating the capability of LLMs in rendering relevance judgments (Faggioli et al., 2023a; Thomas et al., 2023; Zhuang et al., 2023a) and re-scaling (Wadhwa et al., 2023), we explore leveraging the scoring rubric as a contextual guide for LLM prompting. The underlying concept is that the LLM’s output should directly align with calibrated scores according to the given rubric, thereby removing the need for any post-hoc calibration steps. It is important to note that in this method, the LLM itself essentially functions as the ranker, without the integration of an additional neural ranking model. On the TREC dataset, we adopt the prompt method used by Thomas et al. (2023). Conversely, for NTCIR, we adapt our approach to incorporate the specific rubric outlined in the task description of NTCIR-14 WWW-2 (Mao et al., 2019). It is noteworthy that the NTCIR rubric is defined based on scores provided by two annotators,<sup>5</sup> focusing on the quantitative synthesis of annota-

---

<sup>5</sup>For example, “Relevance=3: One annotator rated as highly relevant, one as relevant.”

tions. In contrast, the TREC rubric is more qualitatively oriented, emphasizing the explanation of query-document relationships.<sup>6</sup>

**Category E: Post-hoc calibration of Monte Carlo (MC) sampling LLM predictions.** In this method, we use the LLM as a zero-shot classifier to determine whether a given query and document pair is relevant (denoted as 1) or not (denoted as 0). To mitigate the bias inherent in the most probable generation and to minimize instances of tied scores, which complicate the derivation of rankings, we sample the LLM’s responses 20 times for each input and calculate the average of these scores. Subsequently, we employ Platt scaling (Platt, 2000) to these averaged scores and refine the parameters using the training set. The prompt we use is similar to that of Zhuang et al. (2023a), with one significant modification: we instruct the LLM to output either “relevant” or “nonrelevant” in lieu of “yes” or “no”. This alteration stems from our observation that the LLM exhibits a strong prior towards generating affirmative responses such as “Yes, I can help you with this request...”, which could potentially skew the predictions. By specifying the terms “relevant” and “nonrelevant”, we aim to reduce this bias and achieve more accurate relevance predictions

**Category F: Training NLE-based neural rankers on calibration data.** Building upon our methods for generating and aggregating natural language explanations (NLEs) for query-document pairs, as discussed in Section 6.2, we proceed to fine-tune a BERT model (not fine-tuned for retrieval<sup>7</sup>) to take meta NLEs and yield scale-calibrated ranking scores. In the scenario of the conditional explanation approach (Section 6.2.2.2), where each input is represented with two meta NLEs, one for relevance and one for non-relevance, our method involves an additional processing step. Specifically, we concatenate the [CLS] hidden states obtained from encoding

---

<sup>6</sup>For example, “[3] Perfectly relevant: Document is dedicated to the query, it is worthy of being a top result in a search engine.”

<sup>7</sup>We experimented with using a retrieval-fine-tuned BERT to initialize the ranker that takes NLEs as inputs, but found it to perform significantly worse than general purpose BERT weights.

both NLEs. This concatenated representation then feeds into an additional linear layer, which is responsible for transforming these combined hidden states into a final ranking score. This approach allows the model to integrate insights from both relevance perspectives.

### 6.3.4 Implementation Details

For all experiments that involve LLMs, we employ the LLaMA2-13B-Chat model (Touvron et al., 2023), hosted locally through vLLM<sup>8</sup> and using quantized weights<sup>9</sup>, operated on an A100 GPU. In this study, our neural rankers are based on BERT (Devlin et al., 2019). We initialize these rankers with weights fine-tuned on MS MARCO, referred to as monoBERT<sup>10</sup>, or with weights without retrieval-oriented fine-tuning, simply denoted as BERT<sup>11</sup>. For fine-tuning on scale calibration data, we use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of  $3 \times 10^{-6}$ . The training is conducted over a maximum of 10 epochs, selecting the best model based on validation set loss. To mitigate the impact of randomness due to the limited dataset size in terms of query numbers, each experiment is run with 5 different random seeds. The metrics reported are averaged across these five runs and the statistical significance is determined using t-tests with Bonferroni correction at the 95% confidence level. In relation to the components and hyper-parameters used in Algorithm 2, we employ ROUGE-L (Lin, 2004b) as the text similarity function  $\mathcal{S}$  following Quach et al. (2023). The similarity threshold  $\lambda$  is set to 0.35, with the sampling budget  $k_l$  (maximum number of responses) fixed at 20 and  $k_s$  (maximum number of sentences in the meta NLE) at 30.

---

<sup>8</sup><https://github.com/vllm-project/vllm>

<sup>9</sup><https://huggingface.co/TheBloke/Llama-2-13B-chat-AWQ>

<sup>10</sup><https://huggingface.co/veneres/monobert-msmarco>

<sup>11</sup><https://huggingface.co/bert-base-uncased>

Table 6.2: Ranking and scale calibration performance of baseline methods and our approaches on the TREC dataset. Statistically significant improvements over “Platt Scaling monoBERT” are marked with †.

Cat.	Method	Ranking		Calibration		
		nDCG	nDCG@10	CB-ECE(↓)	ECE(↓)	MSE(↓)
A	Uncalibrated monoBERT	0.799	0.494	1.205	0.320	0.773
B	Post-hoc + monoBERT	0.799	0.494	1.141	0.125	0.684
C	Calibration fine-tune monoBERT	0.776	0.422	1.093	0.221	0.721
	Calibration fine-tune BERT	0.738	0.327	1.253	0.266	0.785
D	LLM prompting w/ rubrics	0.786	0.457	1.000 <sup>†</sup>	1.246	2.137
E	Post-hoc + MC Sampling LLM	0.790	0.473	1.165	0.145	0.673
F	Literal Explanation + BERT	0.815 <sup>†</sup>	0.529 <sup>†</sup>	0.996 <sup>†</sup>	<b>0.067<sup>†</sup></b>	<b>0.602<sup>†</sup></b>
	Conditional Explanation + BERT	<b>0.822<sup>†</sup></b>	<b>0.534<sup>†</sup></b>	<b>0.862<sup>†</sup></b>	0.428	0.832

As outlined in Section 6.1, we by default use calibrated softmax as the loss function when fine-tuning neural ranking models on calibration data. Detailed explorations of the impact of various training objectives on our methods are provided in Section 6.4.2.

## 6.4 Experimental Results and Analysis

In this section, we provide answers and analysis with regard to the four research questions raised in Section 6.3.

### 6.4.1 Utilities of Natural Language Explanations

The central research question of this study is to determine if NLEs generated by LLMs enhance the calibration and ranking performance of neural rankers (**RQ-1**). We present the main evaluation results in Table 6.2 and 6.3, categorizing each method according to the classifications established in Section 6.3.3 for clear distinction.

We find that methods utilizing NLEs yield statistically significant improvements in scale calibration comparing to using raw query-document inputs, exhibiting lower CB-ECE values compared to post-hoc calibrating monoBERT (Category B) and calibration fine-tuning (Category C) across both datasets. Regarding ranking perfor-

Table 6.3: Ranking and scale calibration performance of baseline methods and our approaches on the NTCIR dataset. Statistically significant improvements over “Platt Scaling monoBERT” are marked with †.

Cat.	Method	Ranking		Calibration		
		nDCG	nDCG@10	CB-ECE(↓)	ECE(↓)	MSE(↓)
A	Uncalibrated monoBERT	0.735	0.337	1.757	0.799	1.824
B	Post-hoc + monoBERT	0.735	0.337	1.624	0.457	1.462
C	Calibration fine-tune monoBERT	0.696	0.268	1.843	0.709	1.874
	Calibration fine-tune BERT	0.727	0.285	1.756	0.546	1.416
D	LLM prompting w/ rubrics	0.728	0.328	<b>1.294</b> †	1.194	2.773
E	Post-hoc + MC Sampling LLM	0.736	<b>0.364</b> †	1.677	0.472	1.540
F	Literal Explanation + BERT	<b>0.742</b>	0.340	1.534†	0.355†	1.330†
	Conditional Explanation + BERT	0.720	0.322	1.405†	<b>0.257</b> †	<b>1.290</b> †

mance, our NLE-based methods significantly surpass the baseline “calibration fine-tune BERT,” which shares the same training data and initial weights, with improvements of up to 11.4% in nDCG and 63.3% in nDCG@10 on the TREC dataset. Intriguingly, these methods even exceed the performance of monoBERT (fine-tuned on MS MARCO) on TREC, indicating that LLM-generated explanations provide valuable insights for document ranking. On the NTCIR dataset, the conditional explanation approach slightly underperforms monoBERT in terms of ranking, whereas the literal explanation approach still showcase a significant performance improvement over monoBERT. In summary, RQ-1 can be answered affirmatively that LLM-generated NLEs significantly enhance the scale calibration of neural rankers, often maintaining or even boosting ranking performance in most scenarios.

#### 6.4.2 Consistency across Different Training Objectives

Yan et al. (2022) advocate for addressing the scale calibration of LTR models using calibrated loss functions. Our work, however, shifts focus towards generalizable content understanding for a more complex task: ranking raw texts as opposed to numeric features. To assess the efficacy of our method across various optimiza-

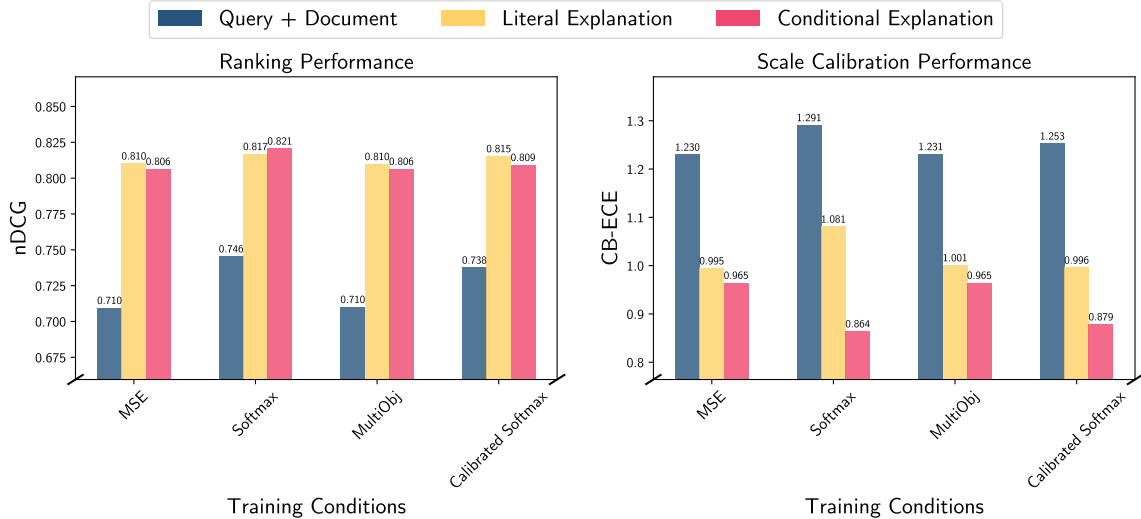


Figure 6.3: Ranking and scale calibration performance of the baseline (neural ranker taking query and documents) and NLE-based approaches on TREC, using four different optimization objectives. NLE-based approaches consistently yield better ranking (left) and calibration (right) performance.

tion objectives, we evaluate it using mean square error (MSE, i.e., regression loss), uncalibrated listwise softmax (Softmax), a multi-objective combination of MSE and Softmax (MultiObj), and calibrated listwise softmax (Calibrated Softmax), all explored in prior research. As a baseline, we fine-tune neural rankers with calibration data (Category C) using query and document inputs. This forms a direct comparison to our NLE-based methods. Outcomes from this analysis are shown in Fig. 6.3. Our findings indicate that the NLE-based approaches consistently outperform traditional neural models processing raw text queries and documents across all four ranking objectives. This result highlights the effectiveness and distinct nature of our scale calibration methods, as well as their versatility and robustness, marking significant improvements over approaches focused solely on calibrated loss functions.

### 6.4.3 Ablations on Natural Language Explanations

To deepen our understanding of the dynamics between LLM-generated NLEs and their impact on scale calibration and ranking, we conducted additional experiments on

Table 6.4: The effect of different types of natural language explanations and selection strategies on the ranking and scale calibration performance of neural rankers.

Explanation	Selection	nDCG	CB-ECE( $\downarrow$ )
Literal	Most Probable	0.789	1.093
	Aggregate MC	<b>0.815</b>	<b>0.996</b>
	Oracle	0.883	0.801
Conditional	Most probable	0.797	0.895
	Aggregate MC	<b>0.822</b>	<b>0.862</b>

the TREC dataset with various types of NLEs. Following the methodology outlined in Section 6.2.3, we tested the aggregation of multiple MC samples of explanations into a single meta NLE (referred to as **aggregate MC**). This method aims to mitigate bias and enrich the reasoning and uncertainty captured in the explanations. We established a control group where only the **most probable** explanation was used as input for the neural ranker for comparative analysis. Additionally, we investigated how the LLM’s inherent limitations, specifically its alignment with human annotator judgments (Faggioli et al., 2023a; Thomas et al., 2023), affect the performance of NLE-based neural rankers. In our literal explanation approach, we continuously sample responses from the LLM, which include both predictions and explanations, until they align with the binary relevance judgments provided by annotators. A match is considered successful if the LLM predicts “relevant” for samples with labels 1, 2, or 3, and “nonrelevant” for those with label 0. If alignment is not achieved within 20 samples, we revert to using the most probable but incorrect explanation, denoted as the **oracle** setting. The results of these experimental setups are presented in Table 6.4.

The results decisively show that the aggregate MC method significantly outperforms the use of the most probable explanation in terms of both ranking and scale calibration, across both literal and conditional explanation setups. This finding validates our hypothesis about the benefits of aggregating multiple explanations and confirms the effectiveness of our strategy, highlighting its utility in enhancing model

performance. Furthermore, the observed performance gap between the aggregate MC approach and the oracle explanation setting underscores a considerable potential for improvement in NLE-based neural rankers. This potential hinges on better aligning LLM judgments with human annotations on the binary scale, possibly through techniques like fine-tuning, prompt engineering, or the use of larger, more capable LLMs. Exploring these enhancements remains an avenue for future research.

## 6.5 Summary

We address the challenge of scale calibration for neural ranking models, aiming to align ranking scores with meaningful real-world measures. Our proposed method capitalizes on zero-shot LLMs’ inherent understanding of textual query-document inputs to enhance scale calibration, while preserving or even boosting ranking performance. We employed natural language explanations (NLEs) generated by LLMs for ranking, demonstrating that our approach surpasses established baseline methods in terms of ranking and calibration metrics and shows consistency across various training objectives. Additionally, we outlined strategies to mitigate the inherent uncertainty in LLM outputs through effective aggregation.

Despite the effectiveness of using zero-shot LLMs, there is potential for improvement through more sophisticated strategies like few-shot prompting (Wei et al., 2022a), instruction tuning (Zhu et al., 2024), and the use of LLMs with more advanced reasoning capabilities. Future research could also focus on increasing the efficiency of NLE generation through techniques like distillation (Gu et al., 2023; Shridhar et al., 2023). Furthermore, enhancing the reliability of explanations (Ye and Durrett, 2022b) represents another promising avenue for developing better calibrated rankers.

## CHAPTER 7

### CONCLUSIONS AND FUTURE WORKS

#### 7.1 Conclusions

In this dissertation, we delve into the development and application of methodologies aimed at enhancing the accessibility, interpretability, and effectiveness of information retrieval systems through the innovative use of explanatory elements.

A principal application of explanations within information retrieval systems is to significantly boost interpretability, thereby increasing user trust. We specifically examine a unique type of explanation—aspect-like information for underspecified queries—and enhance it to incorporate context-awareness. This enhancement involves detailing the relevance of documents in relation to the given query and other documents in the system. Since ranked lists inherently prioritize the comparative relevance of documents, it is crucial that the explanations align with this focus. This is achieved through the development of the LiEGe model (Listwise Explanation Generator), which modifies the encoder and decoder of Transformer-based sequence-to-sequence models to produce individual predictions while referencing other inputs in the batch.

Moreover, we extend the concept of extracting aspects from documents for explanatory purposes, transitioning from a purely post-hoc explanatory approach to an interpretable information retrieval framework that actively promotes search result diversification. Previous approaches to search result diversification either did not integrate the concept of aspects, making them non-explainable, or they relied on an external, un-optimized system to acquire such aspects, resulting in less effective rank-

ings. Our proposed framework, DUB (Diversification Using Bottlenecks), addresses this conflict between interpretability and ranking performance by using an aspect extractor that summarizes query-related information into latent embeddings. These embeddings are optimized for downstream neural rankers that leverage them.

One significant challenge in both tasks is data scarcity; robust explanations of query-document relevance in typical web search settings are limited and costly to obtain. To overcome these challenges, we adopt a transfer learning strategy, pretraining a large-scale task model on a corpus of open-domain data through meticulously designed tasks. This allows our pretrained models to rapidly adapt to specific tasks such as generating explanations and diversifying search results, even with minimal data available for finetuning. Our extensive experiments demonstrate the effectiveness of our model design and strategic use of transfer learning.

In addition to improving interpretability, another less anticipated but equally valuable application of explanations is enhancing the effectiveness of ad-hoc information retrieval systems. This has become more feasible with the surge of large language models (LLMs), addressing the broader issue of data scarcity in information retrieval, which is not solely related to explanations.

Our initial contributions focus on automatic data augmentation, a well-known technique for addressing data scarcity, which we have applied to synthetic query generation in the context of IR. We argue that prior works leveraging LLMs for query generation did not fully utilize their potential and often overlooked critical characteristics of useful training data for neural ad-hoc IR models. To address this, we developed CQG (Contrastive Query Generation), a more challenging task that requires LLMs to produce a query from two documents, designating one as a hard negative. To facilitate this task, we use explanations in prompts to break it down into smaller steps.

Additionally, we explore an alternative approach to data augmentation — leveraging the reasoning capabilities of LLMs through natural language explanations to simplify complex tasks. We focus on the scale calibration of neural ranking models, i.e., ensuring that neural ranking models produce effective ranking scores that adhere to a meaningful scale individually. Essentially, we transition these tasks from ranking textual documents to ranking natural language explanations. Our experiments demonstrate that these techniques significantly advance the tasks.

## 7.2 Future Works

This dissertation establishes a robust foundation for future research in numerous directions within the field of information retrieval, setting the stage for further innovations and enhancements in the accessibility, interpretability, and overall effectiveness of information retrieval systems. Building on the findings presented, several promising directions for future research are clear.

### 7.2.1 Expanding the Scope of Context-Aware Explanations

We have developed tasks and methodologies for context-aware explanation of document relevance, which we consider particularly suited for information retrieval systems due to the comparative nature of relevance and ranking in IR. However, our research scope, which focuses on aspect-like explanations for English ad-hoc retrieval aiming for qualities like relevance and diversity, remains somewhat limited.

One promising direction for future exploration is to address queries beyond the underspecified ones, including factoid and non-factoid questions. This expansion is necessary because not all search engine queries are underspecified, and the adaptability of context-aware explanations must be tested across all possible query types to be genuinely effective. Future research may need to explore an ontology of query types and investigate the most optimal explanation methods for each category through user

studies. It might also be feasible to develop a universal explanation model capable of handling various types of queries.

Another direction to extend the current research is to look beyond English ad-hoc IR. It would be advantageous for the designed approaches to adapt to different languages, supporting monolingual non-English IR, cross-lingual IR, and eventually multilingual IR. Furthermore, considering different modalities applicable to search, such as images, videos, and audio—and potentially a mixture of these modalities, such as in product searches—would significantly enhance search accessibility for a broader range of users globally and cater to a wide array of IR-related applications. Advances in these areas would truly broaden the impact of search result explanations across diverse user groups and applications.

### **7.2.2 Incorporating User Feedback**

Our efforts towards improving the quality and accuracy of explanations generated for search results are mainly evaluated using automatic metrics. The potential perception from users have been overlooked. Incorporating user evaluations and feedback mechanisms into the explanation process presents a valuable avenue for enhancing the relevance and quality of explanations in information retrieval systems. By integrating real-time user interactions, evaluations, and preferences, explanations can be dynamically refined to better meet user needs and improve system transparency.

User evaluations are essential in assessing the effectiveness of explanations provided by information retrieval systems. These evaluations can be structured as surveys or interactive prompts where users rate the clarity, usefulness, and relevance of explanations following their search interactions. This feedback is crucial for identifying aspects of the explanation process that may require refinement.

In addition to direct evaluations, analyzing user interaction patterns with search results and explanations provides deep insights into user preferences and information

needs. This data can be used to personalize explanations, making them more relevant to individual users' contexts and backgrounds. Personalized explanations not only enhance user satisfaction but also increase the likelihood of users trusting and relying on the system for future information needs.

To effectively integrate user feedback, mechanisms such as A/B testing, where different explanation styles are presented to different user groups, can be employed to determine the most effective formats and contents. Additionally, machine learning models can be trained on user feedback data to automatically adjust explanations to align with user preferences and behaviors.

### **7.2.3 Improving Factuality and Faithfulness of Automatically Generated Explanations**

We have developed methods that heavily depend on the capability of large language models (LLMs) to understand human-written explanations, as well as to generate their own explanations with or without human guidance. We have observed that current LLMs often produce hallucinated content and fail to deliver consistent outputs that can be leveraged as reliable data to improve information retrieval systems. Additionally, we note the inconsistency of LLMs in generating explanations, which paradoxically, we leverage to solve certain IR tasks.

Improving the factuality and reducing hallucinations in LLMs are critical steps toward enhancing the quality of explanations used in information retrieval systems. These enhancements ensure that the explanations provided by LLMs are accurate, trustworthy, and ultimately more useful to users and IR system development.

Hallucination in LLMs refers to the generation of information that is not supported by the input data or factual knowledge, leading to misleading or incorrect explanations. To address this issue, one approach involves refining the training processes to better align model outputs with verified data sources. This might be achieved through

techniques such as selective data feeding, where only high-quality, fact-checked information is used during the training phase, thereby teaching the model to replicate such standards in its outputs.

Additionally, incorporating mechanisms to verify the factuality of content generated by LLMs before it is used as an explanation is crucial. This could involve cross-referencing generated explanations against trusted databases or employing secondary models trained specifically to evaluate the truthfulness of text. By implementing these verification processes, information retrieval systems can significantly reduce the risk of disseminating inaccurate information.

Another method to improve factuality is to integrate feedback loops within the system, where users can flag non-factual content. This contributes to continuous model training and improvement. Such user inputs are invaluable for adjusting model parameters and training data, refining the model's ability to generate accurate and factual content.

## BIBLIOGRAPHY

- Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., Smucker, M. D., and Wade, C. (2004). Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189.
- Ai, Q., Hill, D. N., Vishwanathan, S. V. N., and Croft, W. B. (2019a). A Zero Attention Model for Personalized Product Search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19*, pages 379–388, New York, NY, USA. Association for Computing Machinery.
- Ai, Q. and Narayanan, R. L. (2021). Model-agnostic vs. Model-intrinsic Interpretability for Explainable Product Search. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 5–15, Virtual Event Queensland Australia. ACM.
- Ai, Q., Zhang, Y., Bi, K., and Croft, W. B. (2019b). Explainable Product Search with a Dynamic Relation Embedding Model. *ACM Transactions on Information Systems*, 38(1):4:1–4:29.
- Alqahtani, S., Lalwani, G., Zhang, Y., Romeo, S., and Mansour, S. (2021). Using optimal transport as alignment objective for fine-tuning multilingual contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3904–3919.
- Anand, A., Lyu, L., Idahl, M., Wang, Y., Wallat, J., and Zhang, Z. (2022). Explainable Information Retrieval: A Survey. arXiv:2211.02405 [cs].
- Askari, A., Aliannejadi, M., Meng, C., Kanoulas, E., and Verberne, S. (2023). Expand, Highlight, Generate: RL-driven Document Generation for Passage Reranking. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10087–10099, Singapore. Association for Computational Linguistics.
- Bahri, D., Tay, Y., Zheng, C., Metzler, D., and Tomkins, A. (2020). Choppy: Cut Transformer for Ranked List Truncation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1513–1516, Virtual Event China. ACM.

- Bai, A., Jagerman, R., Qin, Z., Yan, L., Kar, P., Lin, B.-R., Wang, X., Bendersky, M., and Najork, M. (2023). Regression Compatible Listwise Objectives for Calibrated Ranking with Binary Relevance. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4502–4508, Birmingham United Kingdom. ACM.
- Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., and Wang, T. (2018). MS MARCO: A Human Generated MACHine Reading COMprehension Dataset. arXiv:1611.09268 [cs].
- Bang, S., Xie, P., Lee, H., Wu, W., and Xing, E. (2021). Explaining A Black-box By Using A Deep Variational Information Bottleneck Approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11396–11404.
- Bastings, J. and Filippova, K. (2020). The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In Alishahi, A., Belinkov, Y., Chrupala, G., Hupkes, D., Pinter, Y., and Sajjad, H., editors, *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.
- Bibal, A., Cardon, R., Alfter, D., Wilkens, R., Wang, X., François, T., and Watrin, P. (2022). Is Attention Explanation? An Introduction to the Debate. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, Dublin, Ireland. Association for Computational Linguistics.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.
- Bonifacio, L., Abonizio, H., Fadaee, M., and Nogueira, R. (2022). InPars: Unsupervised Dataset Generation for Information Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22*, pages 2387–2392, New York, NY, USA. Association for Computing Machinery.
- Boytsov, L., Patel, P., Sourabh, V., Nisar, R., Kundu, S., Ramanathan, R., and Nyberg, E. (2023). InPars-Light: Cost-Effective Unsupervised Training of Efficient Rankers. arXiv:2301.02998 [cs].
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Burges, C. (2010). From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11.

- Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 89–96, Bonn, Germany. ACM Press.
- Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.
- Carmel, D., Haramaty, E., Lazerson, A., Lewin-Eytan, L., and Maarek, Y. (2020). Why Do People Buy Seemingly Irrelevant Items in Voice Product Search? On the Relation between Product Relevance and Customer Satisfaction in eCommerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining, WSDM '20*, pages 79–87, New York, NY, USA. Association for Computing Machinery.
- Carterette, B. and Chandar, P. (2009). Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1287–1296.
- Chandradevan, R., Dhole, K. D., and Agichtein, E. (2024). DUQGen: Effective Un-supervised Domain Adaptation of Neural Rankers by Diversifying Synthetic Query Generation. arXiv:2404.02489 [cs].
- Chaudhary, A., Raman, K., Srinivasan, K., Hashimoto, K., Bendersky, M., and Najork, M. (2023). Exploring the Viability of Synthetic Query Generation for Relevance Prediction. arXiv:2305.11944 [cs].
- Chaudhuri, S., Bagherjeiran, A., and Liu, J. (2017). Ranking and Calibrating Click-Attributed Purchases in Performance Display Advertising. In *Proceedings of the ADKDD'17*, pages 1–6, Halifax NS Canada. ACM.
- Chen, J., Zhang, X., Wu, Y., Yan, Z., and Li, Z. (2018). Keyphrase generation with correlation constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4057–4066.
- Chen, W., Ren, P., Cai, F., Sun, F., and de Rijke, M. (2020a). Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 175–184.
- Chen, W.-F., Syed, S., Stein, B., Hagen, M., and Potthast, M. (2020b). Abstractive Snippet Generation. In *Proceedings of The Web Conference 2020, WWW '20*, pages 1309–1319, New York, NY, USA. Association for Computing Machinery.
- Cho, M., Alizadeh-Vahid, K., Adya, S., and Rastegari, M. (2021). Dkm: Differentiable k-means clustering layer for neural network compression. In *International Conference on Learning Representations*.

- Cohen, D., Mitra, B., Lesota, O., Rekabsaz, N., and Eickhoff, C. (2021). Not All Relevance Scores are Equal: Efficient Uncertainty and Calibration Modeling for Deep Retrieval Models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 654–664. arXiv:2105.04651 [cs].
- Craswell, N., Mitra, B., Yilmaz, E., Campos, D., and Voorhees, E. M. (2020). Overview of the TREC 2019 deep learning track. arXiv:2003.07820 [cs].
- Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E. M., and Soboroff, I. (2021). TREC Deep Learning Track: Reusable Test Collections in the Large Data Regime. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2369–2375, Virtual Event Canada. ACM.
- Dai, Z., Xiong, C., Callan, J., and Liu, Z. (2018). 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*, pages 126–134, Marina Del Rey, CA, USA. ACM Press.
- Dai, Z., Zhao, V. Y., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K., and Chang, M.-W. (2022). Promptagator: Few-shot Dense Retrieval From 8 Examples.
- Dang, V. and Croft, B. W. (2013). Term level search result diversification. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 603–612.
- Dang, V. and Croft, W. B. (2012). Diversity by proportionality: an election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 65–74.
- Dar, G., Geva, M., Gupta, A., and Berant, J. (2022). Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*.
- De Cao, N., Izacard, G., Riedel, S., and Petroni, F. (2020). Autoregressive Entity Retrieval.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

- DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. (2020). ERASER: A Benchmark to Evaluate Rationalized NLP Models. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.
- Dieng, A. B., Ruiz, F. J., and Blei, D. M. (2020). Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.
- Dietz, L., Verma, M., Radlinski, F., and Craswell, N. (2017). Trec complex answer retrieval overview. In *TREC*.
- Faggioli, G., Dietz, L., Clarke, C. L. A., Demartini, G., Hagen, M., Hauff, C., Kando, N., Kanoulas, E., Potthast, M., Stein, B., and Wachsmuth, H. (2023a). Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 39–50, Taipei Taiwan. ACM.
- Faggioli, G., Formal, T., Marchesin, S., Clinchant, S., Ferro, N., and Piwowarski, B. (2023b). Query Performance Prediction for Neural IR: Are We There Yet? arXiv:2302.09947 [cs].
- Fernando, Z. T., Singh, J., and Anand, A. (2019). 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, Paris France. ACM.
- Ferraretto, F., Laitz, T., Lotufo, R., and Nogueira, R. (2023). ExaRanker: Explanation-Augmented Neural Ranker. arXiv:2301.10521 [cs].
- Formal, T., Lassance, C., Piwowarski, B., and Clinchant, S. (2021). SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval. Technical Report arXiv:2109.10086, arXiv. arXiv:2109.10086 [cs] type: article.
- Gao, L. and Callan, J. (2021). Condenser: a Pre-training Architecture for Dense Retrieval. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Gao, L., Ma, X., Lin, J., and Callan, J. (2023). Precise Zero-Shot Dense Retrieval without Relevance Labels. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.
- Geva, M., Schuster, R., Berant, J., and Levy, O. (2021). Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

- Grootendorst, M. (2020). Keybert: Minimal keyword extraction with bert.
- Grootendorst, M. (2022). Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.
- Gu, Y., Dong, L., Wei, F., and Huang, M. (2023). Knowledge Distillation of Large Language Models. arXiv:2306.08543 [cs].
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. arXiv:1706.04599 [cs].
- Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64, Indianapolis Indiana USA. ACM.
- Haag, F., Han, Q., John, M., and Ertl, T. (2014). Aspect grid: A visualization for iteratively refining aspect-based queries on document collections. In *GI-Jahrestagung*, pages 655–660.
- Hashemi, H., Zamani, H., and Croft, W. B. (2021). Learning Multiple Intent Representations for Search Queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 669–679, Virtual Event Queensland Australia. ACM.
- Hastie, T. and Tibshirani, R. (1986). Generalized Additive Models. *Statistical Science*, 1(3):297–310. Publisher: Institute of Mathematical Statistics.
- Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526.
- Hofstätter, S., Mitra, B., Zamani, H., Craswell, N., and Hanbury, A. (2021). Intra-Document Cascading: Learning to Select Passages for Neural Document Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1349–1358, Virtual Event Canada. ACM.
- Hofstätter, S., Zlabinger, M., and Hanbury, A. (2020). Interpretable & time-budget-constrained contextualization for re-ranking. pages 1–8. IOS Press. Accepted: 2022-08-04T16:02:08Z.
- Hu, S., Dou, Z., Wang, X., Sakai, T., and Wen, J.-R. (2015). Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 63–72.
- Huang, J., Gu, S., Hou, L., Wu, Y., Wang, X., Yu, H., and Han, J. (2023a). Large Language Models Can Self-Improve. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

- Huang, Z., Yu, P., and Allan, J. (2023b). Improving Cross-lingual Information Retrieval on Low-Resource Languages via Optimal Transport Distillation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1048–1056, Singapore Singapore. ACM.
- Iwata, M., Sakai, T., Yamamoto, T., Chen, Y., Liu, Y., Wen, J.-R., and Nishio, S. (2012a). AspectTiles: tile-based visualization of diversified web search results. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 85–94, New York, NY, USA. Association for Computing Machinery.
- Iwata, M., Sakai, T., Yamamoto, T., Chen, Y., Liu, Y., Wen, J.-R., and Nishio, S. (2012b). Aspectiles: Tile-based visualization of diversified web search results. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 85–94.
- Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. (2022). Unsupervised Dense Information Retrieval with Contrastive Learning. *Transactions on Machine Learning Research*.
- Jain, S. and Wallace, B. C. (2019). Attention is not Explanation. In Burstein, J., Doran, C., and Solorio, T., editors, *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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jeronymo, V., Bonifacio, L., Abonizio, H., Fadaee, M., Lotufo, R., Zavrel, J., and Nogueira, R. (2023). InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. arXiv:2301.01820 [cs].
- Jiang, Z., Tang, R., Xin, J., and Lin, J. (2021). How Does BERT Rerank Passages? An Attribution Analysis with Information Bottlenecks. In Bastings, J., Belinkov, Y., Dupoux, E., Giulianelli, M., Hupkes, D., Pinter, Y., and Sajjad, H., editors, *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 496–509, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jiang, Z., Wen, J.-R., Dou, Z., Zhao, W. X., Nie, J.-Y., and Yue, M. (2017). Learning to diversify search results via subtopic attention. In *Proceedings of the 40th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 545–554.
- Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, Edmonton Alberta Canada. ACM.

- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense Passage Retrieval for Open-Domain Question Answering. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Khattab, O. and Zaharia, M. (2020). ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, Virtual Event China. ACM.
- Kong, W., Khadanga, S., Li, C., Gupta, S. K., Zhang, M., Xu, W., and Bendersky, M. (2022). Multi-aspect dense retrieval. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3178–3186.
- Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. (2019). Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466. Place: Cambridge, MA Publisher: MIT Press.
- Lampinen, A. K., Dasgupta, I., Chan, S. C., Matthewson, K., Tessler, M. H., Creswell, A., McClelland, J. L., Wang, J. X., and Hill, F. (2022). Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*.
- Lavrenko, V. and Croft, W. B. (2017). Relevance-based language models. *SIGIR Forum*, 51(2):260–267.
- Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.
- Lehman, E., DeYoung, J., Barzilay, R., and Wallace, B. C. (2019). Inferring Which Medical Treatments Work from Reports of Clinical Trials. In Burstein, J., Doran, C., and Solorio, T., editors, *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 3705–3717, Minneapolis, Minnesota. Association for Computational Linguistics.
- Leonhardt, J., Rudra, K., and Anand, A. (2023). Extractive Explanations for Interpretable Text Ranking. *ACM Transactions on Information Systems*, 41(4):1–31.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

- Li, S., Chen, J., Shen, Y., Chen, Z., Zhang, X., Li, Z., Wang, H., Qian, J., Peng, B., Mao, Y., et al. (2022a). Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.
- Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., and Chen, W. (2022b). On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*.
- Lin, C.-Y. (2004a). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Lin, C.-Y. (2004b). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.
- Liu, Y. and Lapata, M. (2019). Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081.
- Loshchilov, I. and Hutter, F. (2019). Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs, math].
- Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., and Kalyan, A. (2022). Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.
- Luan, Y., Eisenstein, J., Toutanova, K., and Collins, M. (2021). Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.
- Lucchese, C., Nardini, F. M., Orlando, S., Perego, R., and Veneri, A. (2022). IL-MART: Interpretable Ranking with Constrained LambdaMART. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2255–2259, Madrid Spain. ACM.
- Lundberg, S. M. and Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Ma, J., Slack, D., Ghandeharioun, A., Singh, S., Lakkaraju, H., et al. (2023a). Post hoc explanations of language models can improve language models. *arXiv preprint arXiv:2305.11426*.

- Ma, X., Zhang, X., Pradeep, R., and Lin, J. (2023b). Zero-Shot Listwise Document Reranking with a Large Language Model. *arXiv:2305.02156 [cs]*.
- MacAvaney, S., Macdonald, C., Murray-Smith, R., and Ounis, I. (2021). Intent5: Search result diversification using causal language models. *arXiv preprint arXiv:2108.04026*.
- Mackie, I., Chatterjee, S., and Dalton, J. (2023). Generative Relevance Feedback with Large Language Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23*, pages 2026–2031, New York, NY, USA. Association for Computing Machinery.
- Mackie, I., Dalton, J., and Yates, A. (2021). How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2335–2341, Virtual Event Canada. ACM.
- MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297.
- Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., and Balahur, A. (2018). WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, pages 1941–1942, Lyon, France. ACM Press.
- Mallia, A., Khattab, O., Suel, T., and Tonellotto, N. (2021). Learning Passage Impacts for Inverted Indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1723–1727, Virtual Event Canada. ACM.
- Mao, J., Sakai, T., Luo, C., Xiao, P., Liu, Y., and Dou, Z. (2019). Overview of the NTCIR-14 We Want Web Task.
- Mayfield, J., Yang, E., Lawrie, D., Barham, S., Weller, O., Mason, M., Nair, S., and Miller, S. (2023). Synthetic Cross-language Information Retrieval Training Data. *arXiv:2305.00331 [cs]*.
- Metzler, D. and Bruce Croft, W. (2007). Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274.
- Miao, N., Teh, Y. W., and Rainforth, T. (2023). SelfCheck: Using LLMs to Zero-Shot Check Their Own Step-by-Step Reasoning.
- Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.
- Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting Similarities among Languages for Machine Translation. In *arXiv:1309.4168 [cs]*. *arXiv: 1309.4168*.

- Montebello, M. (1998). Information overload-an IR problem? In *Proceedings. String Processing and Information Retrieval: A South American Symposium (Cat. No.98EX207)*, pages 65–74.
- Murphy, A. H. and Winkler, R. L. (1977). Reliability of Subjective Probability Forecasts of Precipitation and Temperature. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 26(1):41–47.
- Nguyen, T. T. and Luu, A. T. (2022). Improving neural cross-lingual abstractive summarization via employing optimal transport distance for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11103–11111.
- Nogueira, R. and Cho, K. (2019). Passage Re-ranking with BERT. In *arXiv:1901.04085 [cs]*. arXiv: 1901.04085.
- Nogueira, R., Jiang, Z., Pradeep, R., and Lin, J. (2020). Document Ranking with a Pretrained Sequence-to-Sequence Model. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.
- Nogueira, R. and Lin, J. (2019). From doc2query to docTTTTTquery. page 3.
- Nogueira, R., Yang, W., Cho, K., and Lin, J. (2019a). Multi-Stage Document Ranking with BERT. arXiv:1910.14424 [cs].
- Nogueira, R., Yang, W., Lin, J., and Cho, K. (2019b). Document Expansion by Query Prediction. Technical Report arXiv:1904.08375, arXiv. arXiv:1904.08375 [cs] type: article.
- Nye, M., Andreassen, A. J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., et al. (2021). Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.
- Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., and Cheng, X. (2016). Text Matching as Image Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).
- Pang, L., Xu, J., Ai, Q., Lan, Y., Cheng, X., and Wen, J. (2020). Setrank: Learning a permutation-invariant ranking model for information retrieval. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 499–508.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

- Penha, G. and Hauff, C. (2021). On the Calibration and Uncertainty of Neural Learning to Rank Models for Conversational Search. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 160–170, Online. Association for Computational Linguistics.
- Platt, J. (2000). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.*, 10.
- Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 275–281, New York, NY, USA. Association for Computing Machinery.
- Ponte, J. M. and Croft, W. B. (2017). A Language Modeling Approach to Information Retrieval. *ACM SIGIR Forum*, 51(2):202–208.
- Purpura, A., Buchner, K., Silvello, G., and Susto, G. A. (2021). Neural Feature Selection for Learning to Rank. In Hiemstra, D., Moens, M.-F., Mothe, J., Prego, R., Potthast, M., and Sebastiani, F., editors, *Advances in Information Retrieval*, pages 342–349, Cham. Springer International Publishing.
- Qin, X., Dou, Z., and Wen, J.-R. (2020). Diversifying search results using self-attention network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1265–1274.
- Qin, X., Dou, Z., Zhu, Y., and Wen, J.-R. (2023). Gdesa: Greedy diversity encoder with self-attention for search results diversification. *ACM Transactions on Information Systems*, 41(2):1–36.
- Quach, V., Fisch, A., Schuster, T., Yala, A., Sohn, J. H., Jaakkola, T. S., and Barzilay, R. (2023). Conformal Language Modeling. arXiv:2306.10193 [cs].
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Rahimi, R., Kim, Y., Zamani, H., and Allan, J. (2021). Explaining Documents’ Relevance to Search Queries. arXiv:2111.01314 [cs].
- Reddy, R., Bai, H., Yao, W., Suresh, S. C. E., Ji, H., and Zhai, C. (2023). Social Commonsense-Guided Search Query Generation for Open-Domain Knowledge-Powered Conversations. In Bouamor, H., Pino, J., and Bali, K., editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 873–885, Singapore. Association for Computational Linguistics.

- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 1135–1144, New York, NY, USA. Association for Computing Machinery.
- Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., Gatford, M., et al. (1995). Okapi at trec-3. *Nist Special Publication Sp*, 109:109.
- Saad-Falcon, J., Khattab, O., Santhanam, K., Florian, R., Franz, M., Roukos, S., Sil, A., Sultan, M., and Potts, C. (2023). UDAPDR: Unsupervised Domain Adaptation via LLM Prompting and Distillation of Rerankers. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11265–11279, Singapore. Association for Computational Linguistics.
- Sachan, D. S., Lewis, M., Yogatama, D., Zettlemoyer, L., Pineau, J., and Zaheer, M. (2023). Questions Are All You Need to Train a Dense Passage Retriever. arXiv:2206.10658 [cs].
- Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.
- Santos, R. L., Macdonald, C., and Ounis, I. (2010a). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, pages 881–890.
- Santos, R. L., Macdonald, C., and Ounis, I. (2012). On the role of novelty for search result diversification. *Information retrieval*, 15:478–502.
- Santos, R. L., Peng, J., Macdonald, C., and Ounis, I. (2010b). Explicit search result diversification through sub-queries. In *Advances in Information Retrieval: 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings 32*, pages 87–99. Springer.
- Santos, R. L. T., Macdonald, C., and Ounis, I. (2015). Search Result Diversification. *Foundations and Trends® in Information Retrieval*, 9(1):1–90.
- Sarwar, S. M., Addanki, R., Montazerlghaem, A., Pal, S., and Allan, J. (2020). Search result diversification with guarantee of topic proportionality. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, pages 53–60.
- Shridhar, K., Stolfo, A., and Sachan, M. (2023). Distilling Reasoning Capabilities into Smaller Language Models. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7059–7073, Toronto, Canada. Association for Computational Linguistics.

- Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3145–3153. PMLR. ISSN: 2640-3498.
- Singh, J. and Anand, A. (2019). EXS: Explainable Search Using Local Model Agnostic Interpretability. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19*, pages 770–773, New York, NY, USA. Association for Computing Machinery.
- Singh, J. and Anand, A. (2020). Model agnostic interpretability of rankers via intent modelling. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 618–628, Barcelona Spain. ACM.
- Smucker, M. D. and Clarke, C. L. (2012). Time-based calibration of effectiveness measures. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 95–104, Portland Oregon USA. ACM.
- Su, Z., Dou, Z., Zhu, Y., Qin, X., and Wen, J.-R. (2021). Modeling intent graph for search result diversification. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 736–746.
- Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR. ISSN: 2640-3498.
- Tagami, Y., Ono, S., Yamamoto, K., Tsukamoto, K., and Tajima, A. (2013). CTR prediction for contextual advertising: learning-to-rank approach. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, pages 1–8, Chicago Illinois. ACM.
- Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. (2021). BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models.
- Thomas, P., Billerbeck, B., Craswell, N., and White, R. W. (2019). Investigating Searchers’ Mental Models to Inform Search Explanations. *ACM Transactions on Information Systems*, 38(1):10:1–10:25.
- Thomas, P., Spielman, S., Craswell, N., and Mitra, B. (2023). Large language models can accurately predict searcher preferences. arXiv:2309.10621 [cs].
- Tishby, N., Pereira, F. C., and Bialek, W. (2000). The information bottleneck method. arXiv:physics/0004057.
- Tombros, A. and Sanderson, M. (1998). 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 - SIGIR '98*, pages 2–10, Melbourne, Australia. ACM Press.

- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs].
- TREC (2000). Text REtrieval Conference (TREC) data - English relevance judgements. [https://trec.nist.gov/data/reljudge\\_eng.html](https://trec.nist.gov/data/reljudge_eng.html).
- Turpin, A., Tsegay, Y., Hawking, D., and Williams, H. E. (2007). 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, Amsterdam The Netherlands. ACM.
- Turpin, M., Michael, J., Perez, E., and Bowman, S. R. (2023). Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *arXiv preprint arXiv:2305.04388*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Verma, M. and Ganguly, D. (2019). LIRME: Locally Interpretable Ranking Model Explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR’19*, pages 1281–1284, New York, NY, USA. Association for Computing Machinery.
- Wadhwa, M., Chen, J., Li, J. J., and Durrett, G. (2023). Using Natural Language Explanations to Rescale Human Judgments. arXiv:2305.14770 [cs].
- Wang, K., Thakur, N., Reimers, N., and Gurevych, I. (2022a). GPL: Generative Pseudo Labeling for Unsupervised Domain Adaptation of Dense Retrieval. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2345–2360, Seattle, United States. Association for Computational Linguistics.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. (2022b). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. (2023). Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171 [cs].
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022a). Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022b). Chain of thought prompting elicits reasoning in large language models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., and Zhou, D. (2022c). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Wiegrefe, S. and Pinter, Y. (2019). Attention is not not Explanation. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.
- Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280.
- Xie, Y., Wang, X., Wang, R., and Zha, H. (2020). A fast proximal point method for computing exact wasserstein distance. In Adams, R. P. and Gogate, V., editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*. PMLR.
- Xiong, C., Dai, Z., Callan, J., Liu, Z., and Power, R. (2017). End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–64, Shinjuku Tokyo Japan. ACM.
- Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P. N., Ahmed, J., and Overwijk, A. (2020). Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval.
- Yan, L., Qin, Z., Pasumarthi, R. K., Wang, X., and Bendersky, M. (2021). Diversification-aware learning to rank using distributed representation. In *Proceedings of the Web Conference 2021*, pages 127–136.

- Yan, L., Qin, Z., Wang, X., Bendersky, M., and Najork, M. (2022). Scale Calibration of Deep Ranking Models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4300–4309, Washington DC USA. ACM.
- Ye, X. and Durrett, G. (2022a). The unreliability of explanations in few-shot prompting for textual reasoning. *Advances in neural information processing systems*.
- Ye, X. and Durrett, G. (2022b). The Unreliability of Explanations in Few-shot Prompting for Textual Reasoning. arXiv:2205.03401 [cs].
- Yu, H.-T. (2022). Optimize what you evaluate with: Search result diversification based on metric optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10399–10407.
- Yu, P., Mallia, A., and Petri, M. (2024). Improved Learned Sparse Retrieval with Corpus-Specific Vocabularies. In Goharian, N., Tonellotto, N., He, Y., Lipani, A., McDonald, G., Macdonald, C., and Ounis, I., editors, *Advances in Information Retrieval*, pages 181–194, Cham. Springer Nature Switzerland.
- Yu, P., Rahimi, R., and Allan, J. (2022). 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, Madrid Spain. ACM.
- Yu, P., Rahimi, R., Huang, Z., and Allan, J. (2023). Search Result Diversification Using Query Aspects as Bottlenecks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23*, pages 3040–3051, New York, NY, USA. Association for Computing Machinery.
- Zamani, H., Croft, W. B., and Culpepper, J. S. (2018). Neural Query Performance Prediction using Weak Supervision from Multiple Signals. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 105–114, Ann Arbor MI USA. ACM.
- Zamani, H., Lueck, G., Chen, E., Quispe, R., Luu, F., and Craswell, N. (2020). MIMICS: A Large-Scale Data Collection for Search Clarification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20*, pages 3189–3196, New York, NY, USA. Association for Computing Machinery.
- Zelikman, E., Wu, Y., Mu, J., and Goodman, N. (2022). STaR: Bootstrapping Reasoning With Reasoning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.
- Zerveas, G., Cohen, D., Rekabsaz, N., and Eickhoff, C. (2022). Mitigating Bias in Search Results Through Contextual Document Reranking and Neutrality Regularization. page 7.

- Zhan, J., Mao, J., Liu, Y., Zhang, M., and Ma, S. (2020). An Analysis of BERT in Document Ranking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1941–1944, Virtual Event China. ACM.
- Zhang, R., Guo, J., Fan, Y., Lan, Y., and Cheng, X. (2019a). Outline Generation: Understanding the Inherent Content Structure of Documents. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 745–754, Paris France. ACM.
- Zhang, R., Guo, J., Fan, Y., Lan, Y., and Cheng, X. (2019b). Outline generation: Understanding the inherent content structure of documents. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 745–754.
- Zhang, R., Guo, J., Fan, Y., Lan, Y., and Cheng, X. (2020). Query Understanding via Intent Description Generation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1823–1832, Virtual Event Ireland. ACM.
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019c). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.
- Zhang, Z., Rudra, K., and Anand, A. (2021). Explain and Predict, and then Predict Again. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 418–426, Virtual Event Israel. ACM.
- Zhu, Y., Zhang, P., Zhang, C., Chen, Y., Xie, B., Dou, Z., Liu, Z., and Wen, J.-R. (2024). INTERS: Unlocking the Power of Large Language Models in Search with Instruction Tuning. arXiv:2401.06532 [cs].
- Zhuang, H., Qin, Z., Hui, K., Wu, J., Yan, L., Wang, X., and Berdersky, M. (2023a). Beyond Yes and No: Improving Zero-Shot LLM Rankers via Scoring Fine-Grained Relevance Labels. arXiv:2310.14122 [cs].
- Zhuang, H., Qin, Z., Jagerman, R., Hui, K., Ma, J., Lu, J., Ni, J., Wang, X., and Bendersky, M. (2023b). RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23*, pages 2308–2313, New York, NY, USA. Association for Computing Machinery.
- Zhuang, H., Wang, X., Bendersky, M., Grushetsky, A., Wu, Y., Mitrichev, P., Sterling, E., Bell, N., Ravina, W., and Qian, H. (2021). Interpretable Ranking with Generalized Additive Models. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 499–507, Virtual Event Israel. ACM.