RETRIEVAL AUGMENTED REPRESENTATION LEARNING FOR INFORMATION RETRIEVAL

A Dissertation Presented

by

HELIA HASHEMI

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

DOCTOR OF PHILOSOPHY

May 2024

Robert and Donna Manning College of Information and Computer Sciences

RETRIEVAL AUGMENTED REPRESENTATION LEARNING FOR INFORMATION RETRIEVAL

A Dissertation Presented

by

HELIA HASHEMI

Approved as to style and content by:

W. Bruce Croft, Chair

James Allan, Member

David Jensen, Member

Rajesh Bhatt, Member

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

DEDICATION

To Maman.

ACKNOWLEDGMENTS

First and foremost, I want to express my deepest appreciation to my advisor, Bruce Croft. I am grateful for his unwavering support and belief in my abilities. I joined CIIR at a pivotal moment, when Bruce was considering retirement and questioning whether to expand his group, and I had the privilege of becoming the last PhD student he hired. He took a calculated risk on me, an ambitious student without a research background in IR, who aspired to conduct research in a highly competitive lab like the CIIR. I have worked hard to make his bet pay off, and I'll forever owe my career to the opportunities that Bruce offered to me. Bruce's profound insight and vision, appreciation of prior work, dedication, and work ethics even in times of hardship, have continuously impressed me and have had a significant impact on forming my professional character. I deeply appreciate the autonomy and flexibility he granted me throughout my studies.

I would like to thank my committee members, James Allan, David Jensen, and Rajesh Bhatt, for generously dedicating their time from their busy schedules to review this dissertation and provide invaluable feedback.

I also want to express my gratitude to the CIIR staff. Kate Moruzzi's kindness and patience in working with students have ensured seamless experiences for us. Jean Joyce's proactive approach in solving administrative problems has been invaluable. I greatly appreciate Dan Parker's approachability and technical expertise. Lastly, I enjoyed friendly hallway conversations with Glenn Stowell and appreciate his attention to detail in ensuring that we do not lose funding due to carelessly leaving contracts unsigned.

While all members of the lab have been extremely helpful to me, I want to single out Mohammad Aliannejadi for being a good friend and for teaching me how to conduct careful crowdsourcing experiments. I also want to thank Hamed Zamani, Yen-Chieh Lien, Youngwoo Kim, Chen Qu, Keping Bi, Dan Cohen, and Qinyao Ai for making the lab feel like a friendly community. The random chats in the lab and the shared meals we had after long days have made my experience truly enjoyable.

Hamed Zamani deserves his own distinguished mention for the role he played like a "second advisor" for me. His genuine enthusiasm for the field, his excitement when explaining a new idea, and his exquisite knowledge and intellectual ability inspire me every day. He has been a co-author on most papers introduced in this dissertation, and none of my works would have been possible without our countless discussions and his priceless advice.

I want to thank "Naassaa"—a group of my closest friends since high school, including Nakisa, Nazanin, Tina, Paniz, and Reyhane—for staying connected despite the long distance. I also want to acknowledge Tootiya and Statira for their friendship and support. I want to thank all of them for the laughter and comfort they gave me in times of burnout. Without them, the past seven years would have been unbearable for me.

Last, but certainly not least, I want to thank my family. My father, Kazem, always prioritized our education and made significant sacrifices along the way. I'd like to extend a special thank you to my mom, Bahareh. She grew up in a small town in Iran and, at the age of 18, got into a top-tier medical school in the country, becoming the first person in her family to pursue higher education, let alone the first woman. Coming from a traditional background, she married young and had kids soon after. After finishing medical school, many factors played a role that led her to stay at home with us, dedicating plenty of hours to my childhood to ensure we had the kind of opportunities that would help us succeed. However, her motivation

and perseverance never faded. When she put my sister and me in college, she took the university entrance exam again, got accepted into a PhD program for Addiction Medicine, and recently completed her PhD, starting a new career in her mid-50s. For this reason and many more that she and I know but this acknowledgment is not the place to express them, I dedicate this dissertation to her.

I also want to express my gratitude to my little sister, Kimia. Her independence and critical thinking truly inspire me. Although the long years of the PhD took away the chance of seeing her frequently, the time apart taught me how much I lean on her and how much her contagious smile warms my heart.

I thank my aunt, Bita, and my late grandmother, Minoo; their kindness touched my life immensely and had a profound impact on my development growing up.

I also want to express my deepest and strongest appreciation for my husband, Hamed. I thank him for seeing me through the darkest and most vulnerable times and loving me unconditionally. I thank him for being my safest space, my biggest advocate, and a true friend. His genuine optimism and generous heart have been the fuel that I need to get by every day.

Unfortunately, there is no way for me to give enough credit to everyone who deserves it. CIIR has revolutionized my professional and personal life for the better. Bruce's support has opened doors I never dreamed of before, and I found my life-long partner in the CIIR. For these reasons, I am forever grateful and CIIR will have a unique place in my heart forever.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF IIS-1715095, and in part by a Bloomberg Data Science PhD Fellowship. 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

RETRIEVAL AUGMENTED REPRESENTATION LEARNING FOR INFORMATION RETRIEVAL

MAY 2024

HELIA HASHEMI

B.Sc., AMIRKABIR UNIVERSITY OF TECHNOLOGY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor W. Bruce Croft

Information retrieval (IR) is a scientific discipline within the fields of computer and information sciences that enables billions of users to efficiently access the information they need. Applications of information retrieval include, but are not limited to, search engines, question answering, and recommender systems.

Decades of IR research have demonstrated that learning accurate query and document representations plays a vital role in the effectiveness of IR systems. State-of-the-art representation learning solutions for information retrieval heavily rely on deep neural networks. However, despite their effective performance, current approaches are not quite optimal for all IR settings. For example, information retrieval systems often deal with inputs that are not clear and self-sufficient, e.g., many queries submitted to search engines. In such cases, current state-of-the-art models cannot learn an optimal representation of the input or even an accurate set of all representations.

To address this major issue, we develop novel approaches by augmenting neural representation learning models using a retrieval module that guides the model to-

wards learning more effective representations. We study our retrieval augmentation approaches in a diverse set of somewhat novel and emerging information retrieval applications. First, we introduce Guided Transformer—an extension to the Transformer network that adjusts the input representations using multiple documents provided by a retrieval module—and demonstrate its effectiveness in learning representations for conversational search problems. Next, we propose novel representation learning models that learn multiple representations for queries that may carry multiple intents, including ambiguous and faceted queries. For doing so, we also introduce a novel optimization approach that enables encoder-decoder architectures to generate a permutation invariant set of query intents.

Furthermore, we study retrieval-augmented data generation for domain adaptation in IR, which concerns applying a retrieval model trained on a source domain to a target domain that often suffers from unavailability of training data. We introduce a novel adaptive IR task, in which only a textual description of the target domain is available. We define a taxonomy of domain attributes in information retrieval to identify different properties of a source domain that can be adapted to a target domain. We introduce a novel automatic data construction pipeline for adapting dense retrieval models to the target domain.

We believe that the applications of the developed retrieval augmentation methods can be expanded to many more real-world IR tasks.

CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Research in information retrieval (IR) started in the 1950s with automatic analysis, processing, and indexing of text. IR can be broadly interpreted as an interactive process for connecting users with the "right information" at the "right time" to accomplish a task, where interaction can be multi-modal (e.g., using text, speech, and gestures) and connection can be made in multiple ways (e.g., querying, browsing, and recommendation) [219]. Since its early days, IR research has focused on text data. Similarly, this dissertation focuses on text-based applications.

Users have always been in the center of IR research, which is one of the main factors that distinguishes the information retrieval field from database and even artificial intelligence (AI) [30]. One of the earliest and most important applications of IR was in the library, so naturally, in addition to computer scientists, librarians and information scientists have adopted IR research and have made significant contributions to the field. Over time, the way people access information and their expectations from IR systems have significantly changed. The creation of the Internet, the rise of challenges associated with big data, and the momentum gained by applications like web search resulted in many implications in the design of IR systems. Since then, the role of the user in IR has remained strong [9, 30, 66] but the contributions of the computer science community on IR research have become more and more prominent [29]. The contributions of this dissertation also lie on the system side of information retrieval as we study the core models in an IR system.

IR aims to make access to information easier. The context or the medium in which we define information also matters. Thus, the research problems studied in the field have heavily been restricted to different platforms and applications. One of the core problems of the field since the beginning was representation learning. The common practice for representation learning in IR is to use state-of-the-art pre-trained large language models that are fine-tuned for IR tasks [46, 96, 112]. Despite substantial recent progress in representation learning, it remains a major challenge, because each platform or problem brings its own requirements and constraints. For instance, in web search, queries are often short and, consequently, many queries are vague or incomplete. Consider the *ambiguous query* 'TREC'; the real intent of the user by submitting this query may be the 'Texas Real Estate Commission', or in a more relevant context to this dissertation, the 'Text REtrieval Conference'. Furthermore, people may use the same wording to explain different things. If a user issues the query 'migraine', they may look for 'what triggers migraine?', while another user may use it to find information about 'migraine symptoms'. State-of-the-art IR models learn a single representation for each query, however, given that queries often carry multiple meanings (also called query intents), having a single query representation is not optimal. Similar challenges exist in many other applications, such as conversational search where users have not yet expressed all their needs. This dissertation introduces a number of approaches for such scenarios, where the inputs to the IR models are not sufficient.

This dissertation uses retrieval augmentation [214] as a general framework and designs state-of-the-art retrieval augmented models that will serve IR applications such as query representation, conversational search, and domain adaptation as their downstream tasks. In more detail, we design neural models that take advantage of an embedded retrieval module and evaluate our models for core IR applications. This process is depicted in Figure 1.1. It is important to highlight that existing

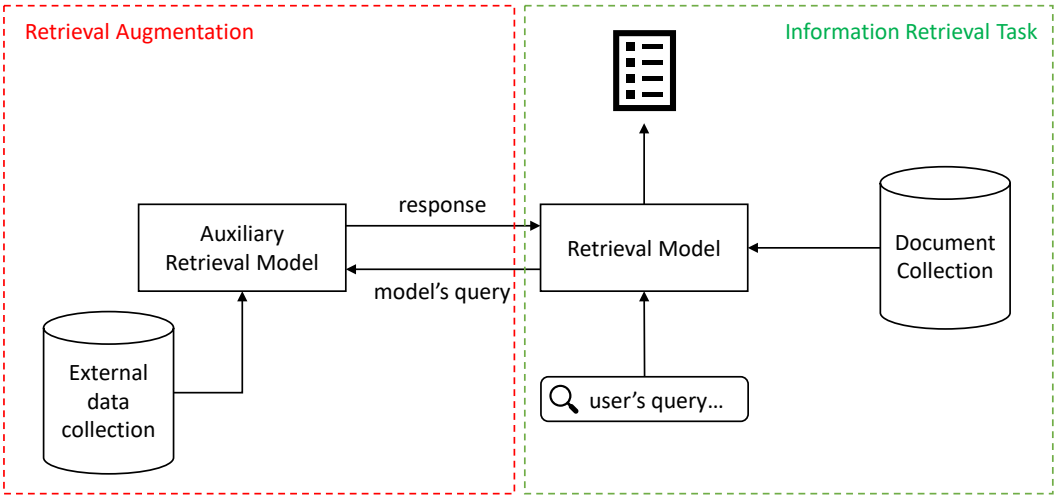

Figure 1.1: Retrieval augmentation for information retrieval.

research has defined the core concepts of IR, such as information need, relevance, feedback, browsing, and evaluation, based on people as the end users of IR systems [30]. However, retrieval augmentation is one of the very few scenarios in which IR system is not serving humans firsthand, but it serves another system. This may require us to revisit the core concepts of IR in the context of retrieval augmentation.

Although retrieval augmented models have recently become a focus of natural language processing and computer vision fields, it has roots in early IR research. Pseudo-relevance feedback (PRF) that was introduced in the 1970s [4, 27] assumes that the top retrieved documents retrieved in response to a query are relevant to the query. Based on this assumption, PRF uses the top retrieved documents for query term weighting and expansion [86, 196, 220]. Even though the idea is close to the concept of retrieval augmentation, it has been studied with completely different terminology and in the context of very few applications, such as query expansion. This dissertation proposes the early attempts to use retrieval-augmented deep learning for IR problems. For more information on retrieval augmentation and its historical context, refer to Chapter 2.

In the rest of this chapter, we discuss the IR applications that we study in this dissertation. We discuss the importance of these applications, the shortcomings that they currently face, and how retrieval augmentation can address them. We pick three diverse and unstudied IR tasks to highlight the universal power and potential of the developed retrieval augmented models for challenging IR tasks. The rest of this section delivers a deep dive into the IR applications that we use in this dissertation and is organized as follows: Section 1.1 focuses on retrieval-augmented models for conversational search as an emerging application in IR industry. Our work on conversational search enables us to learn more accurate representation of information seeking conversations, which results in smarter conversational agents that can ask relevant clarifying questions. Section 1.2 proposes a framework for learning multiple representations for each single input. This has many use cases. In this specific work, we decide to study its applications on representing search queries. We discuss how learning multiple representations can improve the performance of IR systems for queries that are ambiguous or faceted. In Section 1.3 we focus on domain adaptation in IR. We introduce a new category of domain adaptation for neural IR models that has not been studied before. In contrast to previous works, we assume the retrieval model does not have access to the target document collection. Instead, it does have access to a brief textual description that explains the target domain. We define a taxonomy of domain attributes in retrieval tasks to understand different properties of a source domain that can be adapted to a target domain. We propose an automatic data construction pipeline that produces a synthetic document collection, query set, and pseudo relevance labels, given a textual domain description to mimic the target domain data. The model later gets adapted using this data for the target task.

1.1 Retrieval-Augmented Representations for Conversational Search

Conversational search has recently attracted considerable attention as an emerging subarea of information retrieval. The goal of conversational search systems is to address user information needs through multi-turn natural language conversations. This goal is partially addressed in previous work with several simplifying assumptions [18, 32, 140, 146]. Most existing work in conversational search assumes that users always ask a query, and the system only responds with an answer or a ranked list of documents.

There have been some recent efforts to go beyond the "user asks, system responds" paradigm in conversational search by asking clarifying questions from the users, including offline evaluation of search clarification [2], clarifying question generation for open-domain search queries [215], and preference elicitation in conversational recommender systems [19, 157, 224]. Past research in search clarification has shown significant promise in asking clarifying questions. However, utilizing user responses to clarifying questions for improving the search performance has been understudied. In Chapter 3, we propose a model that learns an accurate representation for a given user-system conversation. We focus on the conversations in which the user submits a query, and due to uncertainty about the query intent or the search quality, the system asks one or more clarifying questions to reveal the actual information need of the user. This is one of the many necessary steps that should be taken to achieve an ideal mixed-initiative conversational search system. We propose a neural network architecture that retrieves multiple information sources for learning accurate representations of user-system conversations. We extend the Transformer architecture [176] by proposing *Guided Transformer* – a novel attention mechanism that adjusts the input representation conditioned on multiple retrieved external sources. We train an end-to-end network based on the proposed architecture for two downstream tar-

get tasks: document retrieval in conversational search and next clarifying question selection.

1.2 Learning Multiple Query Intent Representations through Retrieval Augmentation

Representation learning has always played a key role in information retrieval systems. Most retrieval models, including recent neural approaches, use representations to calculate similarities between queries and documents to find relevant information from a corpus. An emerging recipe for achieving state-of-the-art effectiveness in neural IR models involves utilizing large pre-trained large language models (LLMs), e.g., BERT [39] and BART [89], for representing user inquiries and documents [96]. Although these representations benefit from well-designed attention mechanisms and have led to significant performance improvements in many IR and natural language processing (NLP) tasks, they have their own shortcomings in deployment for some tasks. For instance, in query representation learning, which is a core IR problem, the current common practice is to use the query text as the LLM's input and produce a single representation for the query, e.g., see [71, 195]. However, as is widely accepted [154], each query may be associated with multiple intents.[1] We argue that learning a single representation for these queries causes information loss for individual query intents and cannot be semantically inclusive for all query intents. Consequently, it cannot be optimal for many IR applications, including query facet generation, query disambiguation, search result diversification, and clarification in web and conversational search engines.

In Chapter 4, we address this issue by proposing a *general framework* for learning multiple representations for a query such that each representation addresses one of its

[1] In this dissertation, query intent and facet are used interchangeably.

potential intents. Our framework is designed based on a neural retrieval-augmented encoder-decoder architecture and is optimized such that the generic query representations produced by the encoder are transformed into multiple remotely distributed representations, each associated with a query intent. Our framework has applications in a wide range of IR tasks outlined in Chapter 4.

1.3 Domain Adaptation with Description for IR

Domain adaptation is a challenging task in information retrieval, specifically when training data in the target domain is not available or is limited. Given the costly process of gathering training data in different domains, the hope in the task of domain adaptation is to train a retrieval model with existing training data that are available in a source domain, and then find techniques that make the retrieval model adaptable to any given target domain. The traditional setting in this task assumes that the training data and collection is available in the source domain and little or zero training data exists in the target domain in addition to the collection. However, we argue that this is not always a realistic assumption. For many reasons like privacy, access to the target collection may not be feasible. So, we define a scenario in which the retrieval model does not access the target collection for adaptation, but it has access to a textual description that describes the target domain at high level. To the best of our knowledge, this is the first time that the task of domain adaptation is formulated in this setting. In Chapter 5, we define a taxonomy for this new task, and propose a retrieval augmented model that addresses the problem by automatically creating an intermediary collection. Utilizing this intermediary collection, we develop an adaptive retrieval model through pseudo query generation and pseudo labeling, demonstrating significant improvement compared to competitive baselines.

1.4 Contributions

This dissertation focuses on retrieval-augmented models for information retrieval tasks, which are relatively unexplored. We focus on a diverse set of IR applications to evaluate our models. The major published contributions of this dissertation include:

- We design a novel model that learns a better representation of an information seeking conversation. To be more specific, for queries that are ambiguous or faceted, our model is able to retrieve from external sources and utilizes the result list to learn more accurate representations of the conversation. We show that this improves document ranking and next clarification selection in conversational search.

- We show that the go-to practice for learning representation of text, which is using large language models (e.g., BERT and GPT3) is sub-optimal for many IR tasks. We focus on the task of query representation and show that queries that are ambiguous or have multiple facets could benefit from multiple representations such that each representation is associated with one facet of the query. For doing so, we develop NMIR – a novel encoder-decoder model that is able to generate all potential intents behind user's query based on the retrieval results. At the end, we evaluate our model with the accuracy of the generated facets, and how the individual representations affect the performance of the system for retrieval. For efficient training of our model, we suggest an asynchronous optimization approach.

- We further revisit NMIR by addressing one of its shortcomings. NMIR optimization is sequential, meaning that it learns a *list* of multiple representations. However, query representations do not follow a particular order. We address this issue by introducing PINMIR – a model that learns to generate a *set* of text pieces in a permutation invariant manner. This helps us acquire more effective representations for our previous solution to learn multiple representations. This is achieved by (1) introducing a stochastic permutation-invariant optimization approach, and (2)

resetting the positional encoding of Transformer network for each intent description generated by the model.

- We introduce a new category of domain adaptation for information retrieval, where only the description of the target domain is available. We provide benchmark results for this new task and develop a model for automatic creation of a collection from domain description for training an adaptive neural ranking model. We demonstrate that learning an adaptive ranking model from an automatically constructed target collection would lead to significant performance improvements on a wide range of standard IR benchmarks.

CHAPTER 2

RELATED WORK

In this chapter, we review related literature to our work. We discuss how they are relevant but different from what this dissertation offers. In more detail, we start by introducing pseudo-relevance feedback as an early retrieval augmentation approach that has been around for decades. We later review the recent literature on large deep learning models for language modeling and improving them through retrieval augmentation. Next, we provide a brief overview of conversational search and clarification, which has been used in evaluating our models in Chapter 3. We next present the literature on representation learning for search queries, query facet generation, and permutation-invariant representations. These parts are related to the contributions of Chapter 4. We finally review the literature on domain adaptation in neural IR, which is related to our work presented in Chapter 5.

2.1 Pseudo-Relevance Feedback

In information retrieval, pseudo-relevance feedback (PRF), or blind feedback, refers to an approach for updating the query model using the top retrieved documents. This approach assumes the top retrieved documents are relevant to the query, so they can be used for various goals, e.g., query expansion. The main goal of PRF is to improve retrieval performance and this approach has been shown to be highly effective in many retrieval models and tasks [86, 103, 105, 150, 220].

The Rocchio algorithm [150] is one of the earliest relevance feedback methods, which was developed for the vector space retrieval model. This Rocchio algorithm

combines the original query vector with positive and negative feedback vectors which are created using the relevant and non-relevant documents, respectively. Later on, Attar and Fraenkel [4] as well as Croft and Harper [27] proposed to improve the retrieval effectiveness without relevance information using pseudo-relevance feedback. Later, several PRF models have been proposed for the language modeling framework. The comparative analysis of PRF methods done by Lv and Zhai [103] showed that the mixture model [220] and a variant of relevance models (i.e., RM3 [1, 86]) outperform other PRF methods.

The aforementioned methods are based on unigram language models without having access to additional sources of information. Using other information, such as term proximity [104, 110], term topics [204], term dependency [107], and semantic similarity [114, 208, 209], improves pseudo-relevance feedback. There are also a number of learning-based query expansion methods that use thesauruses and external resources [151, 158]. In addition, there has been research to determine which documents can be useful in generating feedback models [35, 55].

Prior work has demonstrated that query expansion with pseudo-relevance feedback leads to significant performance improvement on a number of retrieval tasks. The main reason is that queries are short and consuming the top retrieved documents enables us to obtain better query representations. Even though this dissertation is not directly related to pseudo-relevance feedback, it is strongly influenced by the idea behind it. The retrieval-augmented neural networks came well after the pseudo-relevance feedback models, but they are designed based on the same concept. Retrieval-augmented models use the results of a retrieval model as an input to a network to enrich the representations. Similarly, pseudo-relevance models assume the top retrieved documents are relevant and use them for improving query models.

2.2 Large Language Models

Language models are machine learning models that are trained to predict the likelihood of a sequence of words. Currently, the state-of-the-art approach is to use large transformer-based language models, such as GPT [142] and T5 [144]. An evolving technique for using these models is called "prompting." It refers to using language models to generate text by providing the model with a short text (the prompt) that serves as a starting point for the model's generation. The idea behind prompting is to provide the model with a specific context or task, so that it can generate text that is more focused and coherent.

Prompts can be used for few-shot learning. To be more specific, language models can be fine-tuned for specific tasks using a small amount of task-specific data, such as a few examples or instructions. These models are called instruction-tuned language models. They include T0 [155], InstructGPT [125], and Tk-INSTRUCT [189]. Instruction-tuned models are promising in that they make it possible to fine-tune language models on new tasks with minimal data. InstructGPT [125] argues that it is more effective and truthful than GPT-3 at following user intentions.

In this context, the term "instruction" is distinct from "description" as used in this paper. In previous research, the term "instruction" has been used interchangeably with "intention" and is closely related to the concept of user intent in the field of IR. For example, it was found that if GPT-3 prompted to explain the moon landing to a 6-year-old, it outputs the completion of the prompt text, while InstructGPT generates a more accurate and appropriate response that precisely explains moon landing with simple wording [124]. This is attributed to their training – GPT-3 predicts the next word, while InstructGPT employs techniques such as reinforcement learning from human feedback for fine-tuning the model to better align with user instructions. Other recent research has focused on fine-tuning language models to follow instructions using academic NLP datasets such as FLAN [190] and T0 [155]. However, all these

instruction-based language models are currently limited in their ability to perform complex, multi-step tasks, as opposed to the high-level task-oriented approach used in this study.

Instruction-tuned language models have been effectively applied to various NLP tasks, but have received less attention in the field of IR. This is due to the challenge of casting a retrieval task into the sequence-to-sequence format typically used by these models, as it requires encoding a large corpus of documents. Concurrent to our work, Asai et al. [3] proposed a retrieval method that explicitly models a user's search intent by providing natural language instruction. They concatenated the query with the instruction, encoding it as the query embedding, and then computed the cosine similarity between query and document pairs. In this work, the authors simply concatenated the instruction to the query. However, this approach is limited to handling atomic commands that improve alignment with human intentions, such as "write an answer to this question." These types of instructions are distinct from high-level overviews of complex tasks that require multiple steps to complete.

2.3 Retrieval Augmentation

The basic idea behind retrieval-augmented machine learning models is using a retrieval model to retrieve k documents and employing them in the input of an ML model [214]. The topic is recently in the spotlight due to the challenges it addresses in modern machine learning design, such as "hallucination" in language models.[1] However, as mentioned above, the idea has roots in traditional IR models. Although the terminology used in recent ML literature is vastly different from the IR literature, the core idea has been used in many IR tasks studied since the 1970s. For example, many query expansion approaches use the pseudo-relevance feedback assumption. In

[1] hallucination in language models is defined as generating content that is nonsensical or unfaithful to the provided source content.

addition, open-domain question answering models perform retrieval and then process the retrieved documents (often called the 'reader' model) to produce an answer.

Some works [90] make an analogy between the concept of retrieval augmentation with the memory based architectures [164, 192]. Memory architectures refer to a group of models that benefit from a module that caches useful information for the model and fetches the information as needed. Even though not common, a simple inverted index in a search engine could be considered as a non-parametric memory that the model looks up into whenever a new query comes into the system. Similarly, augmenting the model with the result list of a non-differentiable retrieval system could be considered as accessing to non-parametric memory. Some recent works [47, 88, 90] showed promising results by learning the parameters of a retrieval model and the downstream task at the same time and in an end-to-end manner. The regimen of training these two modules at the same time makes the retrieval-augmented module play as a parametric memory which is the focus of memory architectures.

Despite the idea behind retrieval augmentation being around for many years, the current formulation of the problem emerged after large-scale neural models became mainstream in the fields of ML and NLP, and the problems associate with them, e.g, hallucination, arose.

Some approaches use retrieval components solely for the purpose of optimization, e.g., for producing training data and/or computing loss functions. Thus, the retrieval model will not be used during inference. For instance, ANCE [195] and its extensions [94, 137] are dense retrieval models that iteratively use the model parameters to retrieve documents for producing 'hard' negative samples for training the model. Some early work in weak supervision in IR [38, 211] also inspired the main idea behind retrieval augmentation for optimization.

Among all these, there are two lines of work that are most relevant to the approaches contributed by this dissertation. They include retrieval augmentation for

knowledge-intensive tasks and for language models. In the following subsections, we review these two categories of retrieval-augmented models.

2.3.1 Retrieval Augmentation for Knowledge-Intensive Language Tasks

Recent studies have shown that large language models are able to memorize and generalize knowledge observed in their training set and achieve significant performance on various NLP tasks. However, they still lack grounding in the real-world applications, and their capacity is heavily constrained with task-specific architectures. Access to external knowledge, that may come from a result list returned by a retrieval model, may help with this issue [36, 75, 90, 225]. Open-domain question answering, fact verification, and task-oriented dialogues are examples of knowledge-intensive tasks. Lewis et al. [90] introduced retrieval-augmented generation (RAG) by augmenting a text generator model with the output of a non-parametric retriever that uses maximum inner product search. In open-domain QA, the general approach is to retrieve documents or passages from Wikipedia or the web and then extract an answer from them [63, 141]. Similar retrieval augmentation approaches are also used for multi-hop reasoning and iterative question re-writing [33, 70].

Improving the representations learned by neural models with the help of external resources has been explored in a wide range of tasks. Wu et al. [194] proposed a text matching model based on recurrent and convolution networks that has a knowledge acquisition gating function that uses a knowledge base for accurate text matching. Yang et al. [202] studied the use of community question answering data as an external knowledge base for response ranking in information seeking conversations. They proposed a model based on convolutional neural networks on top of the interaction matrix. More recently, Yang et al. [198] exploited knowledge bases to improve LSTM based networks for machine reading comprehension tasks.

To avoid hallucination in task-oriented dialogues, Thoppilan et al. [172] introduced LaMDA – a language model optimized for chatbots and task-oriented dialogues. The authors collected data from a setting where crowdworkers can use external tools to find factual claims, and trained the model to mimic their behavior.

2.3.2 Retrieval-Augmented Language Models

Khandelwal et al. [69] introduced KNN-LM, a simple extension to language models that uses a retrieval module to find the nearest neighbor tokens given the prefix as query. KNN-LM linearly interpolates the predicted distribution for the next token using distance information from the retrieval mechanism. Similarly, BERT-KNN [68] employs a nearest neighbor algorithm to augment a BERT model to learn better representations for rare facts. Tay et al. [169] formulated the retrieval problem as a sequence-to-sequence task, and proposed training a model that learns the mapping of document content to document identifiers. This can be used to generate relevant document identifiers given a query at inference time. More recently, Borgeaud et al. [12] proposed RETRO for language modeling. They combined a frozen BERT retriever, a differentiable encoder, and a cross-attention mechanism (followed by our contribution in Chapter 3, Guided Transformer) to access the retrieved tokens and make a final prediction based on information from the input and the retrieved items from the database.

2.4 Conversational Search and Question Answering

Conversational search is an emerging application of information retrieval and has attracted considerable attention in recent years. It has roots in early work on interactive information retrieval. For instance, Cool et al. [23] studied how users can have effective interactions with a information seeking system. Later on, Croft and Thompson [28] introduced I^3R, the first interactive IR system with a user modeling

component. Conversational system research in the form of natural language interaction started in the form of human-human interactions [23] or human-system interactions with rule-based models [122, 184]. Some early work also focused on spoken conversations in a specific domain, such as travel [5, 58].

More recently, Radlinski and Craswell [143] introduced a theoretical framework and a set of potentially desirable features for a conversational information retrieval system. Trippas et al. [174] studied real user conversations and provided suggestions for building conversational systems based on human conversations. The recent improvements in neural models has made it possible to train conversation models for different applications, such as recommendation [224], user intent prediction [139], next user query prediction [203], and response ranking [202].

There is also a line of research in the natural language processing community with a focus on conversational question answering [146]. The task is to answer a question from a passage given a conversation history. In this paper, we focus on the conversations in which the system ask a clarifying question from the user, which is fundamentally different from conversational QA literature.

In this dissertation, we choose conversational search as one of the important problems of modern IR and use it as one of the main applications to evaluate our models. In Chapter 3, we explain how the models we developed advance the conversational systems and help them learn a more effective representation of the conversation history.

2.5 Search Clarification

Asking clarifying questions has attracted much attention in different domains and applications. To name a few, Pavel et al. [14] studied user intents and clarification in community question answering (CQA) websites. Trienes and Balog [173] also focused on detecting ambiguous CQA posts, which need further follow-up and clarification.

There is another line of research related to the machine reading comprehension (MRC) task: given a passage, the aim is to generate questions which point out missing information in the passage. Rao and Daumé III [145] proposed a reinforcement learning solution for clarifying question generation in a closed-domain setting. We highlight that most techniques in this area assume that a passage is given, and the model should point out the missing information. Hence, it is different from clarifying the user information needs in IR. Clarification has also been studied in dialog systems [11, 34, 102], computer vision [115], and speech recognition [162]. However, since none of the above-mentioned systems are information seeking, their challenges are fundamentally different from challenges that the IR community faces in this area.

In the realm of IR, the user study done by Kiesel et al. [72] showed that clarifying questions do not cause user dissatisfaction, and in fact, they sometimes increase their satisfaction. Coden et al. [21] studied the task of clarification for entity disambiguation. The clarification format in their work was restricted to a "did you mean A or B?" template. More recently, Aliannejadi et al. [2] introduced an offline evaluation methodology and a benchmark for studying the task of clarification in information seeking conversational systems. They have also introduced a method for selecting the next clarifying question which is used in this paper as a baseline. Zamani et al. [215] proposed an approach based on weak supervision to generate clarifying questions for open-domain search queries. User interaction with clarifying questions has been later analyzed in [217].

A common application of clarification is in conversational recommendation systems, where the system asks about different attributes of the items to reveal the user preferences. For instance, Christakopoulou et al. [19] designed an interactive system for venue recommendation. Sun and Zhang [166] utilized facet-value pairs to represent a conversation history for conversational recommendation, and Zhang et al.

[224] extracted facet-value pairs from product reviews automatically, and considered them as questions and answers.

In Chapter 3 of this dissertation, we study search clarification as one of our applications. We focus on conversational search with open-domain queries which is different from preference elicitation in recommendation, however, the proposed solution can be used for the preference elicitation tasks as well.

2.6 Representation Learning for Search Queries

Learning accurate query representations is a core problem in neural information retrieval. It has applications to query-level tasks, such as query classification [93, 209], query re-writing [57], query auto-completion [16, 129, 187], and query suggestion [37]. It is also an important component in late combination neural ranking models [38, 46], such as DSSM [60], SNRM [213], ColBERT [71], and ANCE [195].

Traditionally, queries were represented based on term occurrences and frequencies [152]. However, these models suffer from the vocabulary mismatch problem: when a concept is represented by different vocabulary in queries and relevant documents. Several studies have tried to address this issue mostly with query expansion and pseudo-relevance feedback [27, 86, 150, 220]. Latent semantic indexing (LSI) is one of the early approaches for learning semantic representation for queries and documents. It calculates a term frequency matrix given a piece of text and uses singular value decomposition for embedding the given text in a semantic space. Alternatively, more recent word embedding models, e.g., word2vec [111] and GloVe [132], learn self-supervised word representations by predicting words given their adjacent words or vice versa in a large text collection. Early attempts to use word embedding models for information retrieval mainly focused on query expansion [82, 208] and document expansion or language model smoothing [43].

Zamani and Croft [209] proposed the first model for deriving query representations from the learned embedding vectors of individual query terms. They introduced a theoretical framework for query representation and showed that a maximum likelihood optimization approach for query representation would lead to averaging the embedding vectors of query terms, if no more information is available. In their follow up work [210], the authors suggested to learn IR-specific word and query embeddings by predicting the words appearing in (pseudo-) relevant documents in response to each query. Diaz et al. [40] alternatively suggested to train word2vec models on local context, i.e., the top retrieved documents in response to the query. Later on, Zhang et al. [221] showed that the relevance-based word embedding of Zamani and Croft [210] can be further trained on clicked documents obtained from a search engine's log, and proposed a generic query representation model that is trained using various implicit feedback signals, e.g., clicks, with multi-task learning. More recently, large-scale contextual embedding models, such as BERT [39], are used to represent queries and documents for a range of IR tasks [96]. These models are often fine-tuned using supervised signals for the downstream task to perform effectively.

All the query representation learning methods pointed out in this section produce a single representation for each query. This single representation can be a single vector for the sequence and/or a single vector per term. However, search queries often carry multiple intents. Therefore, such models theoretically summarize all the query intent representations by their centroid representation. *We believe that neural models should go beyond a single query representation in order to effectively address various information retrieval tasks.* In Chapter 4, we propose solutions for learning multiple representations for search queries to model their various intents.

2.7 Query Facet Extraction and Generation

Search queries can often be characterized by multiple facets, which are implicit or explicit aspects of the query. Implicit facets are often called latent topics. Explicit facets, on the other hand, are words or phrases that represent query aspects. Early work on facet extraction and/or generation [31, 74, 84, 91, 161] focused on applications like e-commerce and digital libraries, where facets can be extracted from existing metadata or taxonomies. These approaches are not extendable to large-scale open-domain settings.

Besides leveraging taxonomies and external resources, some models extract facets by global analysis of the entire search corpus [31, 161]. However, the heterogeneous nature of many search collections, such as the web content, makes such approaches not adoptable [170]. To address this issue, approaches based on local analysis were invented [41, 76, 77]. They extract query facets from the top retrieved documents in the search result list for the query. Notably, Kong and Allan [76, 77, 78] developed a graphical model based approach for facet extraction. They showed that the optimization of their model is an NP-hard problem and thus proposed two approximations (called QFI and QFJ) based on different simplifying assumptions on computing the joint probabilities in the proposed graphical model. Later on, Dou et al. [42] introduced QDMiner that extracts facets with a hybrid approach.

Although query facet generation models do not explicitly learn query representations, they are related to representing different query intents. Therefore, in Chapter 4, we used query facet generation in one of our experiments to evaluate a model for the sake of learning multiple representations for a search query. We compare against the state-of-the-art QFI, QFJ, and QDMiner variations [42, 78] and demonstrate the effectiveness of the proposed solution.

2.8 Permutation-Invariant Representation Learning

Query facets do not often follow a certain ordering. In other words, queries have a *set* of facets. Therefore, a part of the contributions offered in this dissertation is related to set generation where each set member is a piece of text. In more detail, Chapter 4 discusses how we design a set generator network to learn multiple representations for a search query.

Set neural networks are proposed for handling permutation-invariant inputs (i.e., set-input networks) or outputs (i.e., set-output networks). Most existing models focus on set-input problems, where the input of the network is a set of elements. An algorithm designed for set-input problems should satisfy two conditions. First, the model's prediction should remain the same under any permutation of the input (i.e., permutation invariance). Second, such models should take variable input size. Therefore, many existing network architectures such as MLP and RNN cannot be used for set-input networks [116, 123, 168].

One line of work to handle set inputs uses pooling architectures for permutation invariant mapping [101, 159, 160, 163]. Their core idea is to apply the neural function F to each set item individually and apply a pooling permutation-invariant function (e.g., sum or average).

Zaheer et al. [205] discuss the structure of *set pooling* methods and prove that they are a universal approximator for any set function. More recently, attention-based approaches came to the play for the set networks [61, 178, 199]. For instance, Lee et al. [87] proposed Set Transformer which allows the model to encode pairwise or higher order interactions between items in a set.

Set-output networks are less explored. To design a set-output network, the model needs to satisfy two conditions. First, the model must be permutation-equivariant; meaning that the generation of a particular permutation of output should be as

probable as any other permutation. Second, the loss function should be permutation invariant.

Recently, Zhang et al. [223] introduced a model for permutation-equivariant set generation. Following their work, researchers worked on a transformer architecture for predicting a set of object properties [79, 100]. The majority of these approaches study computer vision problems and do not focus on text set generation.

2.9 Domain Adaptation in Neural Information Retrieval

Research in this area can be categorized into two main categories: supervised adaptation and unsupervised adaptation. In supervised (often few-shot) domain adaptation, the assumption is that labeled data is available in the source domain and a (limited) amount of labeled data is available in the target domain, e.g., see Sun et al. [165]. A common approach within this category is transfer learning, which utilizes a pre-trained model from the source domain and fine-tunes it on the target domain using a small set of labeled data. This approach has been shown to improve model performance by allowing the model to learn the specific characteristics of the target domain [39].

Unsupervised domain adaptation assumes that the target document collection is available, but queries and relevance labels are not. Wang et al. [185] proposed a generative pseudo-labeling approach for this scenario. They generated synthetic queries from each document in the corpus and applied a re-ranking based pseudo-labeling approach for each query and document pair. Then, the generated data is used to train a retrieval model. Zhu and Hauff [226] proposed an answer-aware strategy for domain data selection, which selects data with the highest similarity to the new domain. The source data examples were sorted based on their distance to the target domain center, and the most similar examples were chosen as pseudo in-domain data to re-train the question generation model. Additionally, they presented

two confidence modeling methods, namely, generated question perplexity and BERT fluency score, which emphasized labels that the question generation model was more confident about. Recently, Gao et al. [44] introduced a zero-shot dense retrieval model by using a pre-trained generative model to generate hypothetical documents relevant to the query. They used these generated documents as queries and, with the use of a pre-trained dense retrieval model, i.e., Contriever [62], they retrieved documents from the target domain.

CHAPTER 3

RETRIEVAL-AUGMENTED REPRESENTATIONS FOR CONVERSATIONAL SEARCH

Conversational search has recently attracted much attention as an emerging information retrieval (IR) field. The goal of conversational search systems is to address user information needs through multi-turn natural language conversations. This goal is partially addressed in previous work with several simplifying assumptions. For example, the TREC Conversational Assistance Track (CAsT) in 2019 has focused on multi-turn conversational search, in which users submit multiple related search queries [32]. Similarly, conversational question answering based on a set of related questions about a given passage has been explored in the natural language processing literature [18, 140, 146]. However, the existing settings are still far from the ideal *mixed-initiative* scenario, in which both user and system can take any permitted action at any time to perform a natural conversation [2]. In other words, most existing work in conversational search assumes that users always ask a query, and the system only responds with an answer or a ranked list of documents.

Recent conversational information seeking platforms, such as Macaw [207], provide support for multi-turn, multi-modal, and mixed-initiative interactions. There have been recent efforts to go beyond the "user asks, system responds" paradigm by asking clarifying questions from the users, including offline evaluation of search clarification [2], clarifying question generation for open-domain search queries [215],

---

The content of this chapter is largely based on our article published in the proceedings of ACM SIGIR 2020 [51].

and preference elicitation in conversational recommender systems [19, 157, 224]. Past research in search clarification has shown significant promise in asking clarifying questions. However, utilizing user responses to clarifying questions to improve the search performance has been relatively unstudied. In this chapter, we propose a model that learns an accurate representation of user's information need given their conversation with the system. We focus on the conversations in which the user submits a query, and due to uncertainty about the query intent or the search quality, the system asks one or more clarifying questions to reveal the actual information need of the user. This is one of the many necessary steps that should be taken to achieve an ideal mixed-initiative conversational search system.

Motivated by previous research on improving query representation by employing other information sources, such as the top retrieved documents in pseudo-relevance feedback [4, 27, 86], we propose a *retrieval-augmented* neural network architecture that uses multiple information sources for learning accurate representations of user-system conversations. We extend the Transformer architecture [176] by proposing a novel attention mechanism. In fact, the sequence transformation in Transformer networks is guided by multiple external information sources to learn more accurate representations. Therefore, we call our network architecture **Guided Transformer** or GT. We train an end-to-end network based on the proposed architecture for two downstream target tasks: document retrieval in conversational search and next clarifying question selection. In the first target task, the model takes a user-system conversation and scores documents based on their relevance to the user information need. On the other hand, the second task focuses on selecting the next clarifying question that would lead to higher search quality. For each target task, we also introduce an auxiliary task and train the model using a multi-task loss function. The auxiliary task is identifying the actual query intent description for a given user-system conversation. For text representation, our model takes advantage of BERT [39], a pre-

trained language model based on the Transformer architecture, modified by adding a "task embedding" vector to the BERT input to adjust the model for the multi-task setting.

In our experiments, we use two sets of information sources, the top retrieved documents (similar to pseudo-relevance feedback) and the pool of different clarifying questions for the submitted search query. The rationale is that these sources may contain some information that helps the system better represent the user information needs. We evaluate our models using the public Qulac dataset and follow the offline evaluation methodology recently proposed by Aliannejadi et al. [2]. Our experiments demonstrate that the proposed model achieves over 29% relative improvement in terms of MRR compared to competitive baselines, including state-of-the-art pseudo-relevance feedback models and BERT, for the document retrieval task. We similarly observe statistically significant improvements in the next clarifying question selection task compared to strong baselines, including learning to rank models that incorporate both hand-crafted and neural features, including BERT scores.

In summary, the major contributions of this chapter include:

- Proposing a novel attention-based architecture, called Guided Transformer or GT, that learns attention weights from external information sources.
- Proposing a multi-task learning model based on GT for conversational search based on clarification. The multi-task learning model uses query intent description identification as an auxiliary task for training.
- Evaluating the proposed model on two downstream tasks in clarification-based conversations, namely document retrieval and next clarifying question selection.
- Outperforming state-of-the-art baseline models on both tasks with substantial improvements.

3.1 Background: Attention in Neural Networks

Attention is a mechanism for computing weights between different inputs in a neural network. The higher the weight, the higher the attention that the associated representation receives. For example, Guo et al. [45] used IDF as a signal for term importance in a neural ranking model. This can be seen as a form of attention. Later on, Yang et al. [200] used a question attention network to weight the query terms based on their importance. Attention can come from an external source or can be computed based on the representations learned by the network.

The concept of attention has been widely explored in NLP and IR [7, 179]. However, this approach really flourished when the self-attention mechanism proved their effectiveness in a variety of NLP tasks [39, 97, 128, 130, 176]. Transformer networks [176] have successfully implemented self-attention and became one of the major breakthroughs in sequence modeling tasks in natural language processing. For instance, BERT [39] is designed based on multiple layers of Transformer encoders and pre-trained for a masked language modeling task. Self-attention is a specific attention mechanism in which the representations are learned based on the attention weights computed by the sequence tokens themselves. In other words, the sequence tokens decide which part of the sequence is important to emphasize and the representations are learned based on these weights. The original Transformer architecture was proposed for sequence-to-sequence tasks, such as machine translation. The model consists of the typical encoder-decoder architecture. Unlike previous work that used convolution or recurrent networks for sequence modeling, Transformer networks solely rely on the attention mechanism. Each encoder layer, which is the most relevant to this work, consists of a self-attention layer followed by two point-wise feed forward layers. The self-attention layer in Transformer is computed based on three matrices, the query weight matrix W_Q, the key weight matrix W_K, and the value weight matrix W_V. Multiplying the input token representations to these matrices gives us three

matrices Q, K, and V, respectively. Finally the self-attention layer is computed as:

$$Z = \text{softmax}(\frac{Q \times K^T}{\sqrt{d}})V \tag{3.1}$$

where d is the dimension of each vector. This equation basically represents a probability distribution over the sequence tokens using the $softmax$ operator applied to the query-key similarities. Then the representation of each token is computed based on the linear interpolation of values. To improve the performance of the self-attention layer, Transformer repeats this process multiple times with different key, query, and value weight matrices. This is called multi-head self-attention mechanism. At the end, all Zs for different attention heads are concatenated as the output of multi-head self-attention layer and fed into the point-wise fully connected layer.

3.2 Motivation and Problem Formulation

In conversational search systems, users pursue their information needs through a natural language conversation with the system. Therefore, in case of uncertainty in query understanding or search quality, the system can ask the user a question to clarify the information need. A major challenge in asking clarifying questions is utilizing user responses to the questions for learning an accurate representation of the user information need.

We believe that the user-system conversation is not always sufficient for understanding the user information need, or even if it is sufficient for a human to understand the user intent, it is often difficult for the system, especially when it comes to reasoning and causation. For example, assume a user submits the query "migraines", and the system asks the clarifying question "Are you looking for migraine symptoms?" and the user responds "no!". Although negation has been studied for decades in the Boolean information retrieval and negative feedback literature [26, 131, 153, 188], it is

still difficult for a system to learn an effective representation for the user information need.

In the above example, if external information sources cover different intents of query, the system can learn a representation similar to the intents other than "symptoms". We present a general approach that can utilize multiple different information sources for better conversation representation. In our experiments, we use the top retrieved documents (similar to the pseudo-relevance feedback assumption [4, 27]) and all clarifying questions for the query as two information sources. Future work can employ user interaction data, such as click data, and past user interactions with the system as external sources.

Let $Q = \{q_1, q_2, \cdots, q_n\}$ be the training query set, and $F_{q_i} = \{f_{1q_i}, f_{2q_i}, \cdots, f_{nq_i}\}$ denote the set of all facets associated with the query q_i.[1] In response to each query submitted by the user, a number of clarifying questions can be asked. Each conversation in this setting is in the form of $< q_i, c_1, a_1, c_2, a_2, \cdots, c_t, a_t >$, where c_i and a_i respectively denote the i^{th} clarifying question asked by the system and its answer responded by the user. The user response depends on the user's information need. The goal is to learn an accurate representation for any given conversation $< q_i, c_1, a_1, c_2, a_2, \cdots, c_t, a_t >$. The learned representations can be used for multiple downstream tasks. In this chapter, we focus on (1) document retrieval and (2) the next clarifying question selection.

Document Retrieval In this task, each training instance consists of a user-system conversation with clarification, i.e., $< q_i, c_1, a_1, \cdots, c_t, a_t >$, a document from a large collection, and a relevance label.

Next Clarifying Question Selection In this task, each training instance includes a user-system conversation with clarification (similar to above), a candidate clarifying

[1]The approaches are also applicable for ambiguous queries. Therefore, assume F_{q_i} contains all aspects of the query.

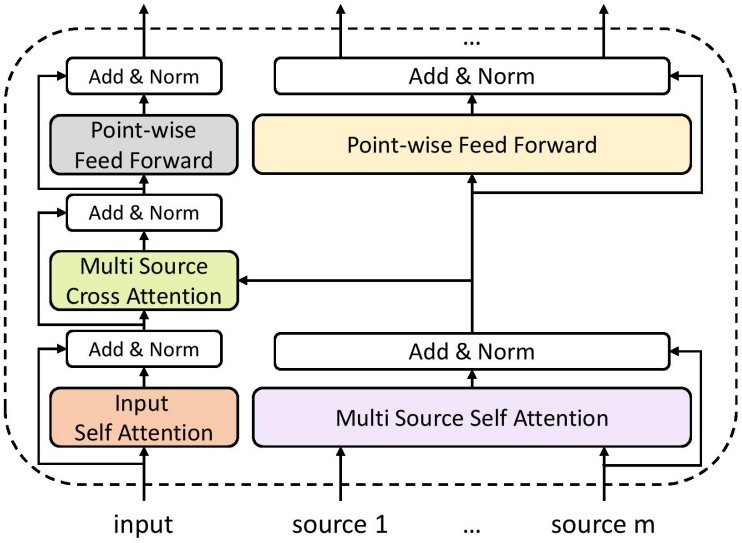

Figure 3.1: The architecture of a Guided Transformer layer.

question c and a label associated with c. The label is computed based on the search quality after asking c from the user.

Our model takes advantage of multiple information sources to better represent each user-system conversation. Let $S_{q_i} = \{s_{1q_i}, s_{2q_i}, \cdots, s_{mq_i}\}$ denote a set of external information sources for the query q_i. Each s_{jq_i} is a text. We later explain how we compute these information sources in our experiments. Note that the term "external" here does not necessary mean that the information source should come from an external resource, such as a knowledge graph. The term "external" refers to any useful information for better understanding of the user-system conversation that is not included in the conversation itself.

3.3 Guided Transformer

The architecture of each Guided Transformer (GT) layer is presented in Figure 3.1. The inputs of each GT layer is an input sequence (e.g., the user-system conversation) and m homogeneous textual information sources. Each source is a set of sequences. We first feed the input sequence to a self-attention layer, called "input self-attention".

This is similar to the self-attention layer in Transformer (see Section 3.1). In the self-attention layer, the representation of each token in the input sequence is computed based on the weighted linear interpolation of all token representations, in which the weights are computed based on the similarity of the query and key vectors, normalized using the softmax operator. See Equation 3.1 for more details. We also apply a self-attention layer to the source representations. In other words, the "multi-source self-attention" layer looks at all the sources and based on their similarity increases the attention on the tokens similar to those frequently mentioned in different source sequences. Based on the idea in residual networks [56], we add the input representations of each self-attention layer with its output and apply layer normalization [6]. This is also the standard technique in the Transformer architecture [176].

In the second stage, we apply attention from multiple external sources to the input representations, i.e., the "multi-source cross attention" layer. In this layer, we compute the impact of each token in external sources on each token in the input sequence. Let t_i and $t_j^{(k)}$ respectively denote the i^{th} input token and the j^{th} token in the k^{th} source ($1 \leq k \leq m$). The output encoding of t_i is computed as follows:

$$\vec{t_i} = \sum_{k=1}^{m} \sum_{j=1}^{|s_k|} p_{\text{ca}} \left(t_j^{(k)} | t_i \right) \vec{v}_{t_j^{(k)}} \tag{3.2}$$

where s_k denotes the k^{th} external source and the vector $\vec{v}_{t_j^{(k)}}$ denotes the value vector learned for the token $t_j^{(k)}$. The probability p_{ca} indicates the cross-attention weight. Computing this probability is not straightforward. We cannot simply apply a softmax operator on top of key-query similarities because the tokens come from different sources with different lengths. Therefore, if a token in a source has a high cross-attention probability, it would dominate the attention weights, which is not desired. To address this issue, we re-calculate the above equation using the law of total probability and the Bayes rule as follows:

$$\vec{t_i} = \sum_{k=1}^{m} \sum_{j=1}^{|s_k|} p\left(t_j^{(k)} | s_k, t_i\right) p\left(s_k | t_i\right) \vec{v}_{t_j^{(k)}} \tag{3.3}$$

In Equation 3.3, $p\left(t_j^{(k)} | s_k, t_i\right)$ denotes the attention weight of each token in source s_k to the token t_i, and $p\left(s_k | t_i\right)$ denotes the attention weight of the whole source s_k to the input token. This resolves the length issue, since $p\left(t_j^{(k)} | s_k, t_i\right)$ is normalized for all the tokens inside the source k, and thus no token in other sources can dominate the weight across all multi-source tokens.

The cross-attention probability $p\left(t_j^{(k)} | s_k, t_i\right)$ is computed by multiplying the key vectors for the tokens in the source s_k to the query vector of the token t_i, normalized using the softmax operator. To compute the attention probability $p\left(s_k | t_i\right)$, we take the key vector for the first token of the s_k sequence. The rationale is that the representation of the start token in a sequence represents the whole sequence [39]. Therefore, the multi-source cross-attention layer can be summarized as follows in a matrix form:

$$\sigma\left(\frac{Q' \times K'^{T}_{\texttt{[CLS]}}}{\sqrt{d}}\right) \sum_{k=1}^{m} \sigma\left(\frac{Q \times K_k^T}{\sqrt{d}}\right) V_k \tag{3.4}$$

where K_k is the key matrix for the k^{th} external source, Q is the query matrix for the input sequence, $K'_{\texttt{[CLS]}}$ is the key matrix for the first token of all sources, Q' is another query matrix for the input sequence (using two separate of query weight matrices), V_k is the value matrix for the k^{th} source, and d is the dimension of the vectors. The function σ is the softmax operator to transform real values in the $[-\infty, \infty]$ interval to a probabilistic space. Similar to the Transformer architecture, both self-attention and multi-source cross-attention layers are designed based on multi-head attention, which is basically repeating the described process multiple times and concatenating the outputs. For the sake of space, we do not present the math for multi-head attention and refer the reader to [176]. Finally, the multi-source cross-attention is followed by

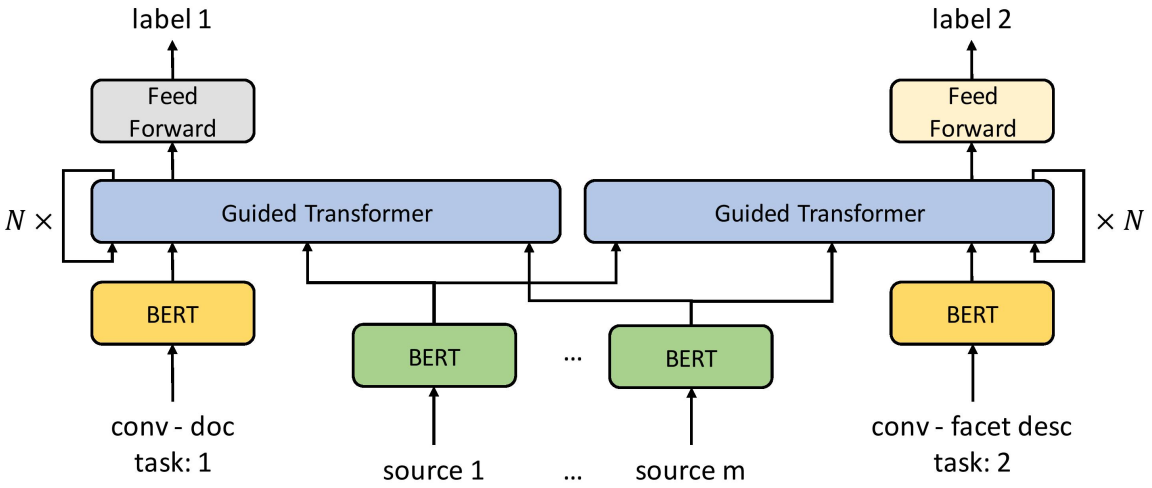

Figure 3.2: The high-level end to end architecture of the model trained using multi-task learning, where the first task is the target task (e.g., document ranking) and the second one is an auxiliary task that help the model identify the user information need from the user-system conversation. Same colors mean shared weights between the networks.

residual and layer normalization. Therefore, multiple GT layers can be stacked for learning more complex representations.

As shown in Figure 3.1, the last step in the multi-source attention layer is a point-wise feed forward network. This is similar to the last step of the Transformer architecture and consists of two point-wise feed forward layers with a ReLU activation in the first one. This feed forward network is applied to all tokens. A final residual and layer normalization produces the output of each multi-source attention layer. The input and output dimension of multi-source attention layers are the same.

If there are different data with multiple instances as source signals, the model can be simply extended by adding more cross-attention layers, one per each data.

3.4 End to End Modeling and Training

The end to end architecture of the model is presented in Figure 3.2. As depicted in the figure, we use BERT for text representation. BERT [39] is a large-scale network

based on Transformer which is pre-trained for a language modeling task. BERT has recently proven to be effective in a wide range of NLP and IR tasks, including question answering [39], passage re-ranking [120, 126], query performance prediction [50], and conversational QA [140]. The coloring in Figure 3.2 shows shared parameters. In other words, we use the same initial parameters for all the models, however, the parameters for BERTs with different colors are different and they are fine-tuned for accurate representation of their inputs. The output of external source representations and input representations are then fed to N Guided Transformer layers introduced above. N is a hyper-parameter. The representation for the start token of the input sequence (i.e., `[CLS]`) is then fed to a feed forward network with a single linear layer to predict the label. Note that for reusing BERT parameters in both tasks (the yellow components) we modified the BERT inputs, which is described later in this section.

As shown in the figure, we train our model using multi-task learning. The first task is the downstream target task (either document retrieval or next clarifying question selection) and the second task is an auxiliary task to help the model better learn the representation to identify the user information need. Note that we can train the model for three tasks (both target tasks and the auxiliary task), but due to GPU memory constraints we limit ourselves to two separate pairs of tasks.

Task 1: The Target Task (Document Retrieval or Next Clarifying Question Selection) For the first task, we concatenate all the interactions in the conversation and separate them with a `[SEP]` token. For the document retrieval task, we concatenate the obtained sequence with the document tokens. For the next clarifying question selection task, we concatenate the obtained sequence with the next clarifying question. Therefore, the input of BERT for the document retrieval task is `[CLS] query tokens [SEP] clarifying question tokens [SEP] user response tokens [SEP] document`

`tokens [SEP]`.[2] Note that BERT has a maximum sequence length limitation of 512 tokens. Some documents in the collection are longer than this length limit. There exist a number of techniques to reduce the impact of this issue by feeding passages to the network and aggregating the passage scores. Since the focus of the work is on learning accurate conversation representations, we simply truncate the documents that are longer than the sequence limit.

Since we re-use the BERT parameters for the second task, we modify the BERT input by adding a **task embedding** vector. In other words, the model learns a representation for each task (in the multi-task learning setting) and we simply add the token embedding, positional embedding, the segment embedding and the task embedding. The first three embeddings are used in the original BERT model.

Task 2: The Auxiliary Task (Intent Description Identification) The clarifying questions are asked to identify the user information need behind the query. For example, each faceted query has multiple facets, and each facet shows a query intent. Therefore, we used the intent description (or facet description) in the data (see Section 3.2 for more details). Similar to the first task we concatenate the user-system conversation with the intent (or facet) descriptions. The label for this task is to identify whether the given facet description describes the user information need or not. In other words, for each user information need, we take some negative samples for training the network and the goal is to distinguish the correct query intent description. This auxiliary task helps the network adjust the parameters by learning attentions that focus on the tokens related to the relevant facet descriptions. Note that the intent description is only available at the training time and at the inference time we put the second part of the model (i.e., task 2) aside.

[2]This example is only valid for a single clarifying question. The user-system conversation can contain more turns.

Loss Function Our loss function is a linear combination of the target loss function and the auxiliary loss function:

$$L = L_{\text{target}} + \alpha L_{\text{aux}} \tag{3.5}$$

where α is a hyper-parameter controlling the impact of the auxiliary loss function. Each of these loss functions is defined using cross entropy. For the task of document retrieval, the labels are binary (relevant vs. non-relevant), while, for the next clarifying question selection task the labels are real numbers in the $[0, 1]$ interval (which are computed based on the retrieval performance if the question is selected). Note that this loss function for the document ranking is equivalent to pointwise learning to rank. Previous work that uses BERT for text retrieval shows that point-wise BERT re-rankers are as effective as the pair-wise BERT models [120].

3.5 Experiments

In this section, we evaluate the proposed model and compare it against state-of-the-art baselines. First, we introduce the data we use in our experiments and discuss our experimental setup and evaluation metrics. We finally report and discuss the results.

3.5.1 Data

To evaluate our model, we use the recently proposed dataset for asking clarifying questions in information seeking conversations, called **Qulac** [2]. Qulac was collected through crowdsourcing based on the topics in the TREC Web Track 2009-2012 [20]. Therefore, Qulac contains 200 topics (two of which are omitted, because they have no relevant document in the judgments). Each topic has been recognized as either "ambiguous" or "faceted" and has been also used for evaluating search result diversification. After obtaining the topics and their facets from the TREC Web Track

Table 3.1: Statistics of the Qulac dataset.

# topics	198
# faceted topics	141
# Ambiguous topics	57
# facets	762
# facet per topic	3.85 ± 1.05
# informational facets	577
# navigational facets	185
# clarifying questions	2,639
# question-answer pairs	10,277

data, a number of clarifying questions have been collected through crowdsourcing for each topic. In the next step, the authors ran another crowdsourcing experiment to collect answers to each clarifying question based on a topic-facet pair. The relevance information was borrowed from the TREC Web Track. The statistics of this dataset is reported in Table 3.1. According to the table, the average facets per topic is 3.85 ± 1.05, and Qulac contains over 10k question-answer pairs.

3.5.2 Experimental Setup

We use the language modeling retrieval model [136] based on KL-divergence [83] with Dirichlet prior smoothing [220] for the initial retrieval of documents from the ClueWeb collection. The smoothing parameter μ was set to the average document length in the collection. For document indexing and retrieval, we use the open-source Galago search engine.[3] The spam documents were automatically identified and removed from the index using the Waterloo spam scorer[4] [24] with the threshold of 70%.

[3] `http://lemurproject.org/galago.php`

[4] `https://plg.uwaterloo.ca/~gvcormac/clueweb09spam/`

We evaluate the models using 5-fold cross-validation. We split the data based on the topics to make sure that each topic is either in the training, validation, or test set. To improve reproducibility, we split the data based on the remainder (the modulo operation) of the topic ID to 5. Three folds are used for training, one fold for validation (hyper-parameter setting), and one fold for testing. After the hyper-parameters were selected based on the validation set, we evaluate the model with the selected parameters on the test set. The same procedure was used for the proposed model and all the baselines.

We implemented our model using TensorFlow.[5] We optimize the network parameters using the Adam optimizer [73] with the initial learning rate of 3×10^6, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L_2 weight decay of 0.01, learning rate warm-up over the first 5000 steps, and linear decay of the learning rate. The dropout probability 0.1 is used in all hidden layers. The number of attention heads in multi-head attentions is set to 8. The maximum sequence length for each document is set to 512 (i.e., the BERT maximum sequence length). We use the pre-trained BERT-base model (i.e., 12 layer, 768 dimensions, 12 heads, 110M parameters) in all the experiments.[6] The batch size was set to 4 due to memory constraints. The other hyper-parameters, including the parameter α (Equation 3.5) and the parameter N (the number of Guided Transformer layers), were selected based on the performance on the validation set.

3.5.3 Evaluation Metrics

Due to the nature of conversational search tasks, we focus on precision-oriented metrics to evaluate the models. We use mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG) [65] with ranking cut-offs of @1, @5, and @20. We report the average performance across different conversations in the data. We

[5] `https://www.tensorflow.org/`

[6] The pre-trained BERT models are available at `https://github.com/google-research/bert`

Table 3.2: The retrieval performance obtained by the baseline and the proposed models. In this experiment, only one clarifying questions has been asked. † and ‡ indicate statistically significant improvements compared to all the baselines with 95% and 99% confidence intervals, respectively. * indicates statistical significant improvements obtained by MTL compared to the STL training of the same model at 99% confidence interval.

	Method		**MRR**	**nDCG@1**	**nDCG@5**	**nDCG@20**
Baselines	QL		0.3187	0.2127	0.2120	0.1866
	RM3		0.3196	0.2189	0.2149	0.2176
	ERM		0.3291	0.2222	0.2191	0.2208
	SDM		0.3235	0.2267	0.2185	0.2182
	BERT		0.3527	0.2360	0.2267	0.2249
GT Network	**Source**	**Loss**				
	Docs	STL	$0.3710^{\ddagger}$	0.2388	$0.2309^{\dagger}$	0.2284
		MTL	$0.3826^{\ddagger *}$	$0.2407^{\dagger *}$	$0.2376^{\ddagger *}$	$0.2328^{\dagger}$
	CQs	STL	$0.4028^{\ddagger}$	$0.2638^{\ddagger}$	$0.2521^{\ddagger}$	$0.2491^{\ddagger}$
		MTL	$0.4259^{\ddagger *}$	$0.2742^{\ddagger *}$	$0.2626^{\ddagger *}$	$0.2543^{\ddagger *}$
	Docs +CQs	STL	$0.4338^{\ddagger}$	$0.2792^{\ddagger}$	$0.2714^{\ddagger}$	$0.2610^{\ddagger}$
		MTL	$\mathbf{0.4554}^{\ddagger *}$	$\mathbf{0.2939}^{\ddagger *}$	$\mathbf{0.2803}^{\ddagger *}$	$\mathbf{0.2697}^{\ddagger *}$

identify statistically significant improvements using the paired t-test with Bonferroni correction at 95% and 99% confidence intervals (i.e., p-value less than 0.05 and 0.01, respectively).

3.5.4 Results and Discussion

In this section, we discuss our empirical results for the two tasks of document retrieval in conversational search and next clarification selection.

3.5.4.1 Document Retrieval for Conversational Search

In the first set of experiments, we focus on conversations with only one clarifying question. The main reason behind this is related to the way the Qulac dataset was created. The questions in Qulac were generated by people operating in a realistic setting. However, the multi-turn setting is not as realistic as the single turn. There-

fore, we first focus on single clarification in our main experiment and later extend it to multi-turn as suggested in [2].

We compare our model with the following baselines:

- QL: The query likelihood retrieval model [136] with Dirichlet prior smoothing [220]. The smoothing parameter was set to the average document length in the collection. This baseline also provides the first retrieval list for the re-ranking models.

- RM3: A state-of-the-art variant of relevance models for pseudo-relevance feedback [86]. We selected the number of feedback documents from $\{5, 10, 15, 20, 30, 50\}$, the feedback term count from $\{10, 20, \cdots, 100\}$, and the feedback coefficient from $[0, 1]$ with the step size of 0.05.

- ERM: Embedding-based relevance models that extends RM3 by considering word embedding similarities [208]. The ERM parameter range is similar to RM3.

- SDM: The sequential dependence model of Metzler and Croft [108] that considers term dependencies in retrieval using Markov random fields. The weight of the unigram query component, the ordered window, and the unordered window were selected from $[0, 1]$ with the step size of 0.05, as the hyper-parameters of the model. We made sure that they sum to 1.

- BERT: A pre-trained BERT model [39] with a final feed forward network for label prediction. This is similar to the method proposed by Nogueira and Cho [120]. BERT-base (i.e., 12 layer, 768 dimensions, 12 heads, 110M parameters) was used in this experiment, which is similar to the setting in the proposed model. The loss function is cross entropy. The optimizer and the parameter ranges are the same as to the proposed method.

Table 3.3: Relative improvement achieved by GT with Docs+CQs and MTL compared to BERT for positive vs. negative user responses to clarifying question. * indicates statistical significant improvements at 99% confidence interval.

Answer	**% MRR improvement**	**% nDCG@5 improvement**
Positive	24.56%*	19.06%*
Negative	31.20%*	25.84%*

The hyper-parameters and training of all the models were done using 5-fold cross-validation, as described in Section 3.5.2. The same setting is used for the proposed methods. Note that the DMN-PRF model proposed by Yang et al. [202] was developed to use knowledge bases as external resources for response ranking in conversation. However, their model cannot accept long text, such as document level text, and thus we cannot use the model as a baseline. The machine reading comprehension models are all extracting answers from a passage and they require a passage as input, which is different from the setting in conversational search. Therefore, such models, e.g., [140], cannot be used as a baseline either. The BERT model has recently led to state-of-the-art performance in many IR tasks and is considered as a strong baseline for us.

We ran our model with different settings: single-task learning (STL) in which the only objective is the target task (i.e., document ranking) and multi-task learning (MTL) that optimizes two loss functions simultaneously. We also use the top 10 retrieved documents (i.e., Docs) and the top 10 clarifying questions (i.e., CQs) as external sources.

The results are reported in Table 3.2. According to the table, the proposed models significantly outperform all the baselines in nearly all cases. Using the clarifying questions as an external information source leads to a higher retrieval performance, compared to using the top retrieved documents. The reasons for this are twofold. First, the documents are often long and can be more than the 512 maximum se-

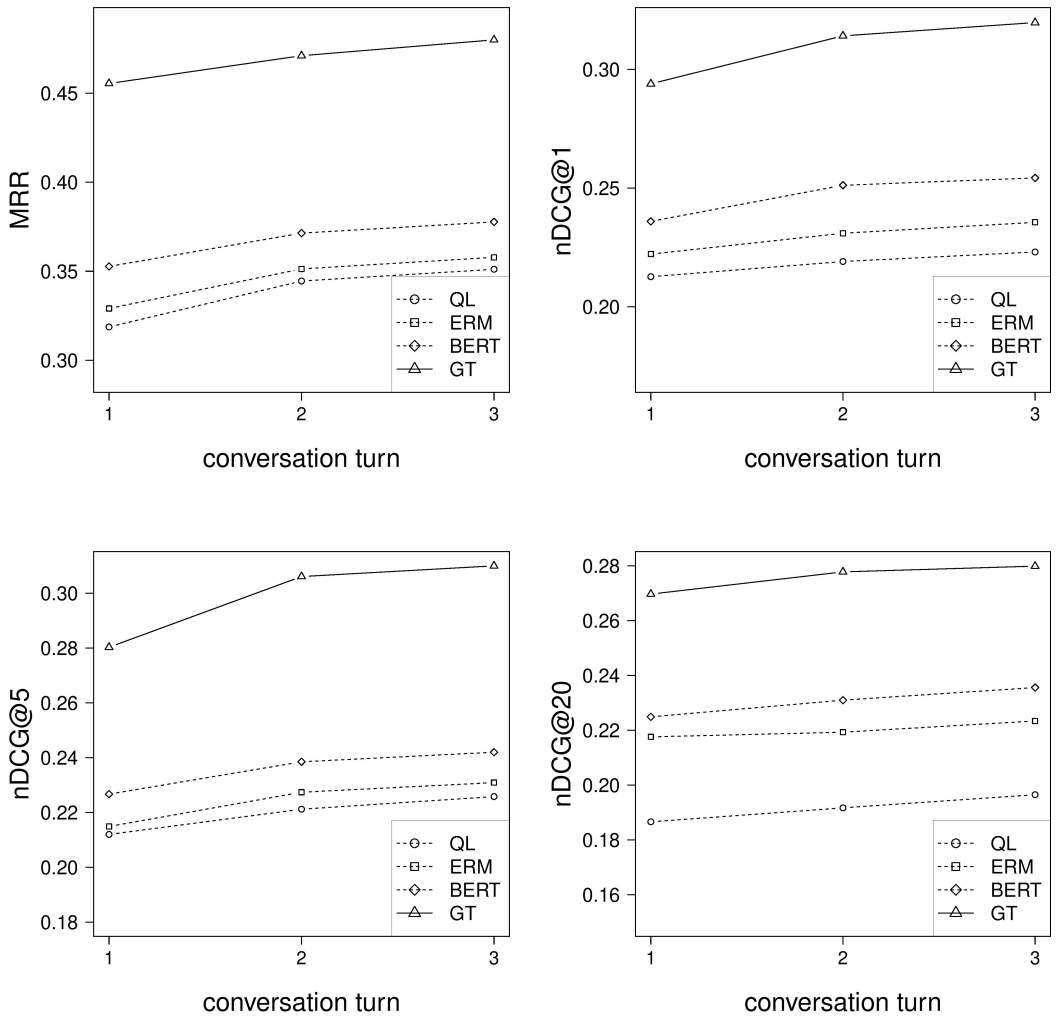

Figure 3.3: The performance of the GT (with Docs+CQs as source and MTL training) compared to the baselines for different conversation turns.

Table 3.4: Relative improvement achieved by GT with Docs+CQs and MTL compared to BERT for user different response length to the clarifying question. * indicates statistical significant improvements at 99% confidence interval.

Answer	**% MRR improvement**	**% nDCG@5 improvement**
Short	33.12%*	26.65%*
Medium	30.83%*	23.93%*
Long	23.41%*	20.36%*

quence length. Therefore, the model truncates the documents and does not utilize all document tokens. Second, the top retrieved documents are not always relevant and non-relevant documents may introduce some noise. On the other hand, the clarifying questions are generated for the particular query and are all relevant. Note that although the clarifying questions in Qulac are human generated, recent methods are able to produce high quality clarifying questions for open-domain search queries. We refer the reader to Zamani et al. [215] for more details on automatic generation of clarifying questions.

According to the results, the multi-task learning setting outperforms the single task learning setting, in all cases. This shows the effectiveness of using an auxiliary task for identifying the correct query intent description. Note that the facet descriptions are only used for training.

Taking both clarifying questions and the top retrieved documents as two information sources leads to the best performance in document retrieval. This shows that these two sources provide complementary information, and the model can effectively utilize this information to better represent the user-system conversations.

Figure 3.3 shows the performance curve as the conversation turn increases. Note that turn i means that i clarifying questions have been asked for each query. For the sake of visualization, we only plot QL, ERM, and BERT as the baselines and the GT with Docs+CQs as external source and MTL as the training setting. According to

the plots, the performance of all models increases with the number of turns. However, the relative improvement in turn 3 compared to turn 2 is less than the improvements observed in turn 2 compared to turn 1. It is not practical to ask too many clarifying questions from users to answer their needs. The plots show it is also not helpful for the system. The proposed method substantially outperforms all the models in terms of all metrics, in all conversation turns.

To better understand the performance of the model, we report the results per different properties of user responses to clarifying questions. In this experiment, we only focus on a single clarifying question. We first identify yes/no questions (based on some simple regular expressions and the question response), and report the relative performance compared to BERT (our best baseline). For the sake of space, we only report the results for MRR and nDCG@5. The improvements for the other metrics also follow a similar behavior. For the proposed model, we focus on GT with the Docs+CQs source and MTL training. The results are presented in Table 3.3. According to the table, GT achieved higher improvements for negative responses. The reason is that for positive responses, the clarifying question already contains some important terms about the user information need, and thus using external information sources leads to smaller improvements. This is true for all metrics, two of which are reported in the table.

We extend our analysis to the response length as well. In this experiment, we divide the test conversations into three equal size buckets based on the user response length to clarifying questions. The results are reported in Table 3.4. According to the table, GT achieves higher improvements for shorter user responses. This is due to the information that is in the responses. In other words, it is easier for BERT to learn an effective representation of user information need if enough content and context are provided, however, for shorter responses, external information sources are more helpful. In both Tables 3.3 and 3.4, the improvements are statistically significant.

Table 3.5: Results for the next clarifying question selection task, up to 3 conversation turns. † and ‡ indicate statistically significant improvements compared to all the baselines with 95% and 99% confidence intervals, respectively. * indicates statistical significant improvements obtained by MTL compared to the STL training of the same model at 99% confidence interval.

Method	**MRR**	**nDCG@1**	**nDCG@5**	**nDCG@20**
OriginalQuery	0.2715	0.1381	0.1451	0.1470
σ-QPP	0.3570	0.1960	0.1938	0.1812
LambdaMART	0.3558	0.1945	0.1940	0.1796
RankNet	0.3573	0.1979	0.1943	0.1804
BERT-NeuQS	0.3625	0.2064	0.2013	0.1862
GT-Docs-STL	0.3784‡	0.2279‡	0.2107†	0.1890
GT-Docs-MTL	**0.3928**$^{\ddagger *}$	**0.2410**$^{\ddagger *}$	**0.2257**$^{\ddagger *}$	**0.1946**$^{\ddagger *}$
Oracle-Worst Question	0.2479	0.1075	0.1402	0.1483
Oracle-Best Question	0.4673	0.3031	0.2410	0.2077

3.5.4.2 Next Clarifying Question Selection

In the second downstream task, we focus on the next clarifying question selection. The task is to select a clarifying question given a user-system conversation. To be consistent with the results presented in [2], we use up to three conversation turns and report the average performance. The quality of models is computed based on the retrieval performance after asking the selected clarifying question from the user. Since the focus is on the next clarifying question selection, we use query likelihood as the follow up retrieval model, similar to the setting described in [2].

We compare our method against the following baselines:

- OriginalQuery: The retrieval performance for the original query without clarifying question. We use this baseline as a term of comparison to see how much improvement we obtain by asking a clarifying question.

- σ-QPP: We use a simple yet effective query performance predictor, σ [133], as an estimation of the question's quality. In other words, for a candidate clarifying question, we perform retrieval (without the answer) and estimate the

Table 3.6: Some examples with a single clarification turn. ΔMRR is compared relative to the BERT performance.

Information need	Find sites for MGB car owners and enthusiasts.
Query	mgb
Clarifying question	are you looking for what mgb stands for?
User response	no
ΔMRR	+78%
Information need	What restrictions are there for checked baggage during air travel?
Query	air travel information
Clarifying question	where are you looking to travel to?
User response	doesn't matter i need information on checked baggage restrictions during air travel
ΔMRR	3%
Information need	What states levy a tax against tangible personal property?
Query	tangible personal property tax
Clarifying question	would you like to find out how much you owe?
User response	no i just want to know which states levy a tax against tangible personal property
ΔMRR	-21%

performance using σ. The clarifying question that leads to the highest σ is selected.

- LambdaMART: We use LambdaMART to re-rank clarifying questions based on a set of features, ranging from the query performance prediction to question template to BERT similarities. The exact definition of feature descriptions can be found in [2].

- RankNet: Another learning to rank model based on neural networks that uses the same features as LambdaMART.

- BERT-NeuQS: A model based on BERT used for clarifying question re-ranking proposed in [2].

The hyper-parameters of these baselines were set using cross validation, similar to the proposed method. The results are reported in Table 3.5. For the proposed method we only use the top retrieved documents as the external information source since this is the most realistic setting for selecting clarifying questions. The proposed model significantly outperforms all baselines, including the recent BERT-NeuQS model. Consistent with the document retrieval experiments, the multi-task learning setting led to higher performance compared to STL. In addition, the achieved performance compared to the original query shows the benefit of asking for clarification.

The table also contains the results for two oracle models, one that always selects the best question, which sets the upper-bound performance for this task on the Qulac data, and the one that always chooses the worst question (i.e., the lower-bound). The best possible performance is still much higher than the one achieved by the proposed solution and shows the potential for future improvement.

3.5.4.3 Case Study

We report three examples with one conversation turn for the document retrieval task in Table 3.6. The ΔMRR was computed relative to the BERT performance achieved by our best model. We report one win, tie, and loss example. In the first example, the user response is "no" and most baselines, including BERT, cannot learn a good representation from such conversation, however, the proposed model can use the top retrieved documents and the other clarifying questions to observe what are then other intents of the query and learn a better representation for the user information need. In the second example, the performance of the proposed model is similar to BERT, and in the last one, GT loses to BERT. The reason is that although the user response is negative, it contains some useful terms related to the information need, which makes it easier for the models to retrieve relevant documents.

The proposed model, however, could not leverage external resources to learn effective representations.

3.6 Summary

In this chapter, we introduced Guided Transformer (GT) by extending the Transformer architecture. GT can utilize external information sources for learning more accurate representations of the input sequence. We implemented GT for conversational search tasks with clarifying questions. We introduced an end to end model that uses a modified BERT representations as input to GT and optimizes a multi-task objective, in which the first task is the target downstream task and the second one is an auxiliary task of discriminating the query intent description from other query intents for the input user-system conversation. We evaluated the proposed model using the recently proposed Qulac dataset for two downstream tasks in conversational search with clarification: (1) document retrieval and (2) next clarifying question selection. The experimental results suggested that the models implemented using GT outperform state-of-the-art baselines in both tasks.

CHAPTER 4

LEARNING MULTIPLE INTENT REPRESENTATIONS FOR SEARCH QUERIES

Neural network approaches have shown promising results in many information retrieval (IR) tasks [46], including but not limited to ad hoc retrieval [45], web search [113], personal search [206], and conversational search [51, 138, 201]. An emerging recipe for achieving state-of-the-art effectiveness in neural IR models involves utilizing large pre-trained language models (LLMs), e.g., BERT [39] and BART [89], for representing user inquiries and documents [96]. Although these representations benefit from well-designed attention mechanisms and have led to significant performance improvements in many IR and NLP tasks, they have their own shortcomings in deployment for some certain tasks. For instance, in query representation learning, which is a core IR problem, the current common practice is to use the query text as the LLM's input and produce a single representation for the query, e.g., see [71, 195]. However, as is widely accepted [154], each query may be associated with multiple intents.[1] We argue that representing these queries using a single representation causes information loss for individual query intents and cannot be semantically inclusive for all query intents. Consequently, it may be optimal for many IR applications, including query facet generation, query disambiguation, search result diversification, and clarification in web and conversational search engines.

The content of this chapter is largely based on our articles published in the proceedings of ACM CIKM 2021 [52] and ACM CIKM 2022 [53].

[1] In this chapter, query intent and facet are used interchangeably.

In this chapter, we address this issue by proposing a **general retrieval-augmented framework** for learning multiple representations for a query such that each representation addresses one of its potential intents.

Our framework, called NMIR, is designed based on a neural encoder-decoder architecture, and is optimized such that the generic query representations produced by the encoder are transformed to multiple remotely distributed representations, each associated with a query intent.

We study both parametric and non-parametric variations of the framework. In the former, the model assumes that the number of representations per query is given, while the latter dynamically identifies the number of representations for each query.

We optimize our framework based on the following hypothesis: if the query encoder can accurately learn multiple query intent representations, therefore the decoder should be able to accurately generate all intent descriptions. On this basis, the training objective in NMIR is to maximize the likelihood of generating the query intent descriptions (or facets). To improve the efficiency of our framework, we introduce an *asynchronous* training strategy in which one process is responsible for model training and another one adjusts the enforcement conditions that obligates the model to generate multiple representations.

NMIR has applications in a wide range of IR tasks reviewed in Section 4.1. We perform extensive experiments for extrinsic evaluation of the model using a query facet generation task. We demonstrate significant improvements compared to competitive baselines using offline evaluation on reusable test collections in addition to manual pairwise comparison with the baseline using three trained annotators.

4.1 Potential Applications

NMIR is a general framework with multiple applications in a wide range of IR tasks. For instance it can be simply used for *abstractive query intent (or facet) gen-*

eration. We use this task in our experiments to demonstrate the quality of learned representations. Another potential application of NMIR would be on *search result diversification*, as multiple query intent representations can help diversify a result list. One can imagine a clear application of NMIR in *exploratory search* tasks, where different representations of the search query can be used by the user to navigate through various aspects of the topic. In *conversational search*, asking clarifying questions has been recognized as an important and challenging task [2]. Multiple query representations can be used for generating and selecting clarifying questions in conversational search systems.

Apart from query representation and its applications, the proposed solution can be potentially adopted for a variety of tasks related to document representation. For instance, according to the scope hypothesis [149], long documents often cover several different topics. Therefore, learning multiple representations for each document can be further investigated using the proposed framework. This will have applications in *document clustering and categorization*. We believe that learning multiple query and document representations together can potentially lead to improvement in *document ranking* too, as the model would be theoretically able to accurately find the closest query intent to the document.

One can even imagine applications of the proposed framework beyond text representation. For instance, in *collaborative recommender systems*, models learn a single representation for each user and item from user-item interaction signals. However, users may have multiple different interests and a single user representation vector may lead to information loss. The proposed framework can be potentially extended to recommender systems by learning a variable number of user representations based on different user interests. This would further lead to recommendation precision improvement. It can be also used for explaining each recommendation. Such technique would also enable users to select what profile representation would they prefer to be

used for the next recommendation, or they can be selected automatically based on the user's situational context [156].

4.2 Task Description and Problem Formulation

Training query representation learning models that are able to produce multiple representations for each search query has not yet been explored. This is a challenging task, especially when the number of representations varies across queries. The task is to learn multiple representations for each search query. We use the top retrieved documents in a search result list in response to the query as a source of evidence to find various intents of the query for representation learning. For training the model, we assume that a textual description of each query intent is available. In Section 4.3.2, we discuss potential solutions on obtaining such descriptions.

Before formalizing the task, we introduce our notation. Let $Q = \{q_1, q_2, \ldots, q_n\}$ be a training query set with n queries, and $D_i = \{d_{i1}, d_{i2}, \ldots, d_{im}\}$ be the top m retrieved documents in response to the query q_i using a retrieval model M. Moreover, let $F_i = \{f_{i1}, f_{i2}, \ldots, f_{ik_i}\}$ denote the set of all textual intent descriptions associated with the query q_i. k_i is the number of query intents and can vary across queries. The task is to learn k_i representations $R_i = \{R_{i1}, R_{i2}, \ldots, R_{ik_i}\}$ for the query q_i, where R_{ij} is the j^{th} representation learned for the query.

4.3 The NMIR Framework

In this section, we describe the proposed NMIR framework, its optimization, and implementation details.

4.3.1 A High-Level Overview

One straightforward solution for the task is using an encoder-decoder architecture that leverages the query q_i (and the top retrieved documents) as the input and

generates multiple query intent descriptions of the query by taking the top k_i most likely predictions, e.g., using beam search. However, previous work in a number of NLP tasks [177, 191] showed that these generations are often synonyms or refer to the same concept, which is in contrast to the goal of our task: learning multiple representations, each associated with a query intent. This solution generates different but semantically similar outputs, which are only related to one query intent. Hence, this approach would not serve the purpose.

Another straightforward solution is to look at the task as a sequence-to-sequence problem, similar to machine translation, and generate all the query intent descriptions concatenated with each other (and separated using a special token). The concern regarding this approach is that different intent representations are not distinguishable in the last layer of the model. In addition, most existing effective text encoding models are not able to represent long sequences of tokens, such as a concatenation of the top m retrieved documents.

The NMIR framework addresses these issues. Let $\phi(\cdot)$ and $\psi(\cdot)$ denote a text encoder and decoder pair, respectively.

For every query q_i in the training set, NMIR assumes that the top retrieved documents D_i are relevant to the query and they may be relevant to different query intents. NMIR assigns each learned document representation to one of the query intent descriptions $f_{ij} \in F_i$ using a document-intent matching algorithm γ:

$$\mathcal{C}_i^* = \gamma\left(\phi(d_{i1}), \phi(d_{i2}), \ldots, \phi(d_{im}), \phi(f_{i1}), \phi(f_{i2}), \ldots, \phi(f_{ik_i})\right)$$

where $\mathcal{C}_i^* = \{C_{i1}^*, C_{i2}^*, \ldots, C_{ik_i}^*\}$ is a set of document sets. Each C_{ij}^* is a set of documents from D_i that are assigned to f_{ij} by γ.

NMIR then transforms the encoded general query representation to its intent representations through a query intent encoder ζ. In more detail, the representation

for the j^{th} query intent is obtained using $\zeta(q_i, C^*_{ij}; \phi)$. The implementation details of components ϕ, ψ, γ, and ζ are presented in Section 4.3.2.

NMIR's training for a mini-batch b is based on a gradient descent-based minimization of $\mathcal{L}(b) = \frac{1}{|b|} \sum_{q_i \in b} L(q_i)$, where $L(q_i)$ is defined as follows:

$$L(q_i) = \frac{1}{k_i} \sum_{j=1}^{k_i} L_{\text{CE}}(f_{ij}, \psi(\zeta(q^*_{ij}, C^*_{ij}; \phi)))$$

where q^*_{ij} = "$q_i f_{i1} f_{i2} ... f_{ij-1}$ ¡mask¿...¡mask¿" is a concatenation of the query string, the first $j-1$ intent descriptions, and $k_i - j$ mask tokens. There is a special separation token between each of these strings. Therefore, $L(q_i)$ basically calculates the loss for generating each textual intent description, given the associated cluster C^*_{ij} and the encoded query text plus the past $j-1$ intent descriptions. This helps the model avoid generating the previous intent representations and learn multiple representations.

In the above loss function, L_{CE} is the cross-entropy loss borrowed from the sequence-to-sequence model [167]:

$$-\sum_{t=1}^{|f_{ij}|} \log p\left(f_{ijt} | \psi(\zeta(q^*_{ij}, C^*_{ij}; \phi)), f_{ij1}, f_{ij2}, \ldots, f_{ijt-1}\right)$$

where f_{ijt} is the t^{th} token in the given intent description f_{ij}.

Inference. Using NMIR at inference time is partly different from the way it is used during training. To be precise, the q^*_{ij}s are constructed differently. At training, they are constructed by concatenating the query and the previous intent descriptions in order to generate the next one. While at inference, we do not have access to the intent descriptions, therefore we should construct q^*_{ij}s based on the model's output. Therefore, for the query q_i, we first feed "q_i $<$ mask $>$ ... $<$ mask $>$" to the model (the number of mask tokens is equal to $|\mathcal{C}^*_i|$) and apply *beam search* to the decoder's output to obtain the first intent description f'_{i1}. We then use the model's output

to iteratively create the input for the next step "$q_i f'_{i1}$ $<$ mask $>$ ... $<$ mask $>$" and repeat this process for $|\mathcal{C}^*_i|$ times. As mentioned earlier, similar to the model training, the reason for including previous outputs is to avoid generating repetitive intent descriptions.

4.3.2 Model Implementation and Training

This subsection describes the detailed implementation of our framework for each of its components. We implemented our model using the PyTorch Lightning platform.[2]

The encoding and decoding components ϕ and ψ. As depicted in Figure 4.1, we use Transformer encoder and decoder architectures for implementing ϕ and ψ, respectively. We initialize their parameters with the pre-trained BART model [89]. BART is a denoising autoencoder for pretraining sequence-to-sequence models. It uses standard Transformer-based encoder-decoder architecture and has been pre-trained based on adding noise to the input text and reconstructing it. In extreme cases, where the input text is corrupted to the extent that there is no information left from the original format, BART is equal to language models. We use the BART's implementation delivered by the HuggingFace's Transformer library [193].[3] In NMIR, the decoder's cross-attention is the output of the intent encoder ζ for each query intent (see Figure 4.1).

The intent encoding component ζ. As shown in Figure 4.1, the intent encoding component $\zeta(q^*_{ij}, C^*_{ij}; \phi)$ is implemented using N' layers of Guided Transformer [51] (see Chapter 3). Guided Transformer is used for influencing an input representation by the *guidance* of some external information. In our case, we use $\phi(q^*_{ij})$ as the input representation and $\phi(d) : \forall d \in C^*_{ij}$ as the external information. In fact, Guided Transformer uses self-attention on the input tokens (the query), self-attention on each

[2] `https://www.pytorchlightning.ai/`

[3] `https://huggingface.co/transformers/`

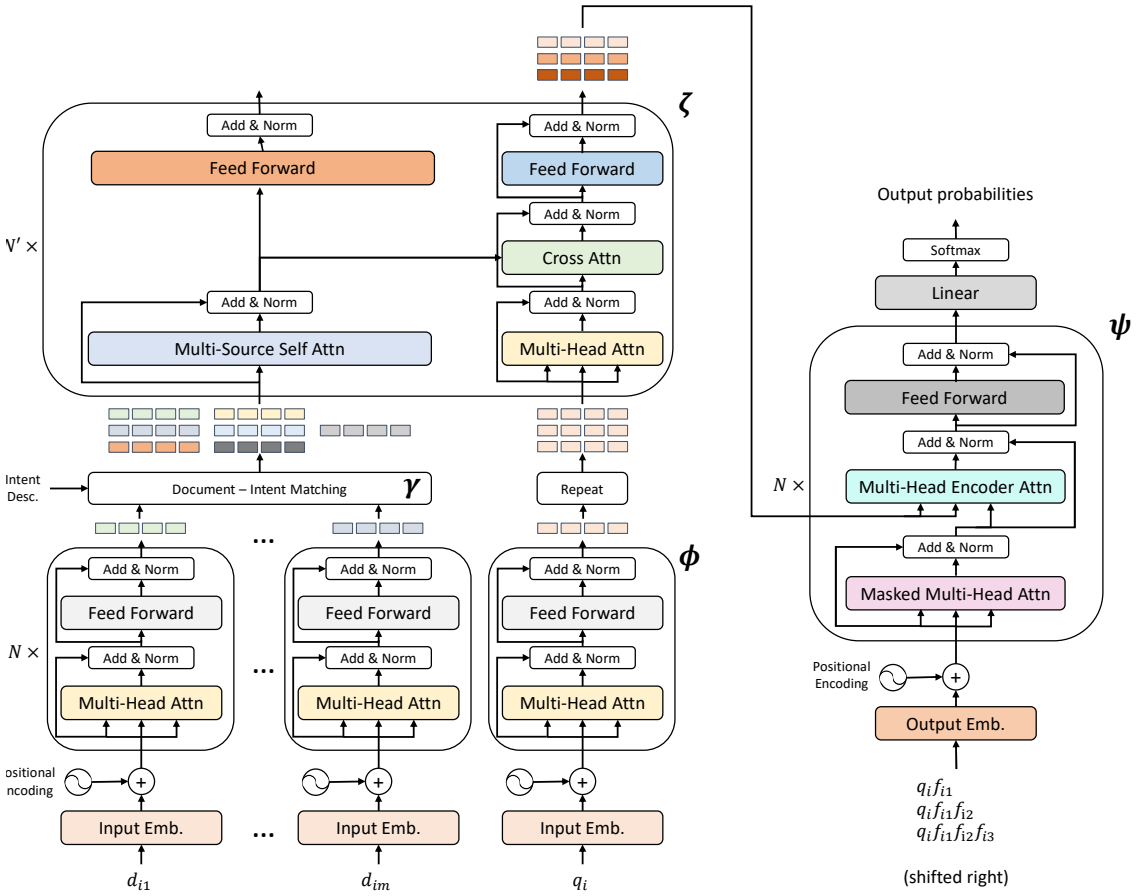

Figure 4.1: The network architecture of NMIR. Same background colors indicate parameter sharing. White background means that the component does not have learnable parameters. The encoder and decoder parameters (ϕ and ψ) are initialized by BART pre-trained parameters [89] consisting of N Transformer layers and are fine-tuned.

external resource (each document in C^*_{ij}), and a cross-attention from the document representations to the query representation. This cross-attention mechanism enables the model transform the generic query representation to a query intent representation.

The document-intent matching component γ. Inspired by work on multi-sense word embedding [92, 117], for document-intent matching based on the encoded representations, we develop an algorithm that clusters the learned representations and assigns each cluster to an intent description. In more detail, NMIR encodes all the top retrieved documents and creates k_i clusters, using a clustering algorithm. Therefore, we have:

$$\mathcal{C}_i, \mathcal{M}_i = \text{cluster}(\phi(d_{i1}), \phi(d_{i2}), \ldots, \phi(d_{im}))$$

where $\mathcal{C}_i = \{C_{i1}, C_{i2}, \ldots, C_{ik_i}\}$ denotes a set of clusters and each C_{ij} contains all the documents in the j^{th} cluster associated with the query q_i. $M_i = \{\mu_{i1}, \mu_{i2}, \ldots, \mu_{ik_i}\}$ is a set of all cluster centroids such that $\mu_{ij} = \text{centroid}(C_{ij})$. In our implementation, we use K-Means [49] for clustering in this step, due to its simplicity and efficiency. K-Means has been successfully used in a number of IR applications [64, 81, 98, 180]. Note that K-Means requires the number of clusters as input. The number of clusters for q_i at the training time is given by the number of intent descriptions (i.e., k_i). However, this value is unknown at inference time. In our experiments, we consider two cases. In the first case, we assume that the number of clusters at test time is equal to a tuned hyper-parameter k^* for all queries. In the second case, we replace the K-Means algorithm by a non-parametric version of K-Means [109]. This algorithm basically starts with creating one cluster based on a minimum document similarity threshold. Once the first cluster is created, the same process would be repeated for the rest of documents that are not yet assigned to any clusters. For more information on non-parametric K-Means, we refer the reader to [109].

The component γ requires a one-to-one assignment between the cluster centroids and the query intents in the training data. The assignment needs to be one-to-

one, since otherwise all clusters may be assigned to a single most dominant query intent, and thus the model would not learn to generate far-flung query representations. Therefore, NMIR uses the following injective surjective function, called the intent identification function $\mathcal{I}$:

$$\mathcal{I}(M_i, F_i) = \arg \max_{M' \in \text{perm}(M_i)} \sum_{j=1}^{k_i} \text{sim}(\phi(f_{ij}), \mu'_j)$$

where perm$(\cdot)$ returns all permutations of a given set and each $M' = [\mu'_1, \mu'_2, \ldots, \mu'_{k_i}]$ denotes a permutation of cluster centroids in M_i. The function sim$(\cdot, \cdot)$ denotes a similarity function. We use inner product to compute the similarity between an intent representation and a cluster centroid. Therefore, let $M_i^* = [\mu_{i1}^*, \mu_{i2}^*, \ldots, \mu_{ik_i}^*]$ be the output of $\mathcal{I}(M_i, F_i)$ and $\mathcal{C}_i^* = \{C_{i1}^*, C_{i2}^*, \ldots, C_{ik_i}^*\}$ be their associated clusters. The component γ returns $\mathcal{C}_i^*$.

Note that the γ is not differentiable and cannot be part of the network for gradient descent-based optimization. Our asynchronous training (presented below) addresses this issue by taking γ out of the optimization process and moving it to an asynchronous process (see Figure 4.2). Another key point is that there is no need to call the function $\mathcal{I}$ at inference time, because the order of the clusters does not matter, while it matters for training as it helps us compute the loss function.

Asynchronous training. As is widely known, the training speed of deep learning models can be greatly improved by using GPUs, mainly due to the huge amount of parallel computation in large-scale neural networks. However, during the training of our model, we observed that the clustering of document representations become an efficiency bottleneck, even after we deploy a K-Means algorithm that runs on GPU. To solve this issue, we consider an asynchronous document encoding and clustering approach depicted in Figure 4.2. In this training approach, we use two GPUs: we save

a snapshot of the encoder parameters (i.e., ϕ) at the beginning of each training step,[4] and compute the document representations for all documents retrieved in response to all training queries. We then use the obtained cluster centroids (M_is) for training the model on the second GPU. While the model is being trained, the first GPU computes the document representations and cluster centroids for the next step. In fact, this approach may not be as effective as synchronous training, because the cluster centroids at each training step is obtained from the model parameters at two previous steps (i.e., as shown in Figure 4.2, the model parameters from step $s-1$ produces the clusters for step $s+1$). However, the efficiency improvement provides enough incentives to consider asynchronous training. We do not have effectiveness comparison between the synchronous and asynchronous training strategies, as training the synchronous model would be impractical on a large dataset.

Training data and setup. Another challenge in training NMIR is related to its training data and especially ground truth intent descriptions. There are multiple ways of automatically creating training data for weak supervision training of the model, for example using query reformulation data or anchor text. In our experiments, we follow a weak supervision solution based on the MIMIC-Click dataset, recently released by Zamani et al. [216].[5] The authors extracted and generated the query intent descriptions by mining and predicting them from the Bing's search query logs. In more detail, the data is created based on query reformulation data with the goal of finding query reformulations that reveal different intents of the query. Since users mostly clarify their intents by adding one or more terms to their original query in a search session, often called query specialization [85], query intents can be predicted by extracting a set of query reformulation triples (q, qq', c) (or $(q, q'q, c)$), which denotes

[4]Note that each training step includes 10000 batches in our experiments.

[5]The MIMICS dataset is available at `https://github.com/microsoft/MIMICS`.

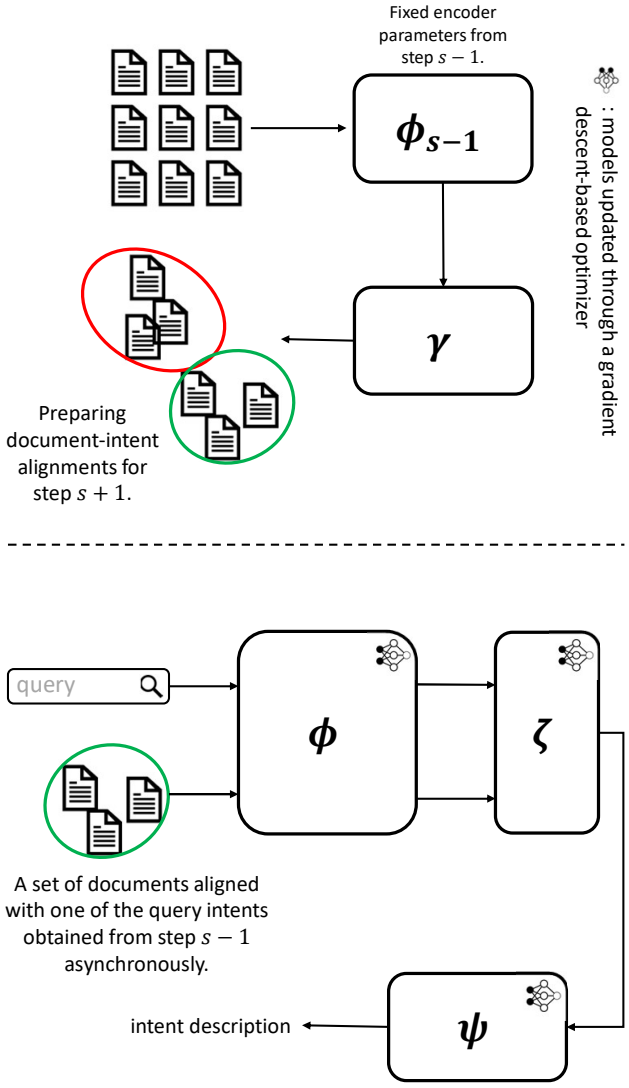

Figure 4.2: The asynchronous training of the NMIR framework. These two steps (above and below the dashed line) are executed on two different GPUs, and the model parameters are only updated in one of the steps, using a gradient descent-based optimizer. ϕ_{s-1} represents the encoder whose parameters are fixed and obtained from a model snapshot at step $s-1$.

that the query q is followed by the query qq' (or $q'q$) in the same search session (i.e., immediate successive queries) with a frequency of c, when it is aggregated over the whole query log data for all users. qq' is the concatenation of q and q', where $|q'| > 0$. Since the mined query reformulations may refer to the same intent, a diversification based approach is used for identifying a diverse set of query intent descriptions [215]. The data consists of over 400,000 unique search queries and 2-5 intent descriptions per query.

In more detail, we use 80% of the MIMICS-Click queries for training and the rest for validation. The validation set is used for hyper-parameter tuning and early stopping. For the top retrieved documents (i.e., D_is), we used the SERP information fetched from the Bing's public web search API by the creators of the MIMICS dataset.[6] In our experiments, we use the document snippets as an accurate textual representation of the retrieved documents.

We used Adam optimizer with a batch size of 8 to train our model. The small batch size was selected due to the GPU memory constraints. We used early stopping based on the loss value on the validation set. The number of Guided Transformer layers was set to three. The learning rate was selected based on the validation loss from the $[1e-6, 5e-5]$ interval. We report the generated facets for a few example queries by NMIR in Table 4.3. The first part of the table includes some successful examples where the model successfully identified the facets of the query, and the second part includes two failure cases. In the first failed query, the model could not distinguish between the word "window" and the windows operating system. As a result, it generated meaningless facet descriptions. The second failed query contains some facet descriptions that may be semantically related to the query but are not

[6]The MIMICS SERP data is available at `http://ciir.cs.umass.edu/downloads/mimics-serp/MIMICS-BingAPI-results.zip`.

coherent. One of the generated facets for this query is even long and grammatically incorrect.

4.4 PINMIR: A Permutation-Invariant Variation of NMIR

Despite its strong performance, NMIR still suffers from some limitations. First, it uses the standard sequence-to-sequence optimization, as a result, it assumes that the query intents are ordered, and it tries to optimize the model to produce intent descriptions in the same order as it appears in the ground truth. Second, NMIR uses a greedy algorithm for assigning each cluster to a ground truth query intent during training. Therefore, the model's performance depends on this heuristic cluster-intent assignment algorithm.

So, in this section, we introduce a permutation-invariant optimization solution for text generation, when each element of the set is a piece of text. We explain our model as a variant of NMIR, where the performance of the model is not sensitive to the order of generated query intent descriptions. In this model, we no longer need the intent-cluster matching algorithm since the order of generated intents does not matter. A side benefit is that sometimes documents address more than one query intent and assigning only one intent to a document would be sub-optimal.

First, we need to define a permutation-invariant loss function for training the model. Common permutation-invariant loss functions include Chamfer loss and Hungarian loss. Chamfer loss is based on Chamfer distance that was first introduced in computer vision [8]. Although it is more efficient, it is not applicable to our task due to the design of decoder for text generation. The reason is that the decoder generates the output token-by-token and the closest ground truth facet is not known until the facet is fully generated. Therefore, we extend the Hungarian loss [80] for text set generation. The proposed loss function for a query q_i is computed as follows:

$$L(\hat{F}_i, F_i) = \min_{F'_i \in \pi(F_i)} L_{CE}(\hat{F}_{ij}, F'_{ij})$$
$$= \min_{F'_i \in \pi(F_i)} \frac{1}{k_i} \sum_{j=1}^{k_i} \sum_{t=1}^{|f'_{ij}|} -\log p(f'_{ijt} | v, f'_{ij1}, f'_{ij2}, \cdots, f'_{ijt-1})$$

where $\pi(F_i)$ denotes all permutations of ground truth intents for query q_i. Therefore, the size of $\pi(F_i)$ is equal to $k_i!$. The loss function L_{CE} is the average sequence-to-sequence loss for generating each facet description, and v denotes the encoder representation. Intuitively, the proposed loss function computes all permutations of ground truth set and considers the one with the minimum loss value, which is the loss value for the closest ground truth ordering to the generated set. Therefore, the original ordering of ground truth text would not impact the loss value.

This loss function can be quite expensive to compute since it requires us to repeat this process for every permutation of the query intents. We propose to use a **stochastic variation** of this loss that instead of iterating over all possible permutations, takes s samples from the permutation set and computes the loss based on the sampled query intent sequences. Our experiments show that the stochastic loss performs comparably to the non-stochastic variation of the loss, which is computationally expensive.

Position Resetting. We highlight that in our task in contrast to the standard assumption in *set networks*, although the order does not matter between the set elements, it matters within each individual element. In other words, the order that the model generates different query intent descriptions does not matter, but it is important that sequence of tokens in each query intent description are generated legitimately, both semantically and syntactically. To help the model capture this concept, we modify the standard decoder architecture in transformer. The decoder generates tokens one-by-one and each token becomes the decoder's input for generating the next token. The standard transformer decoder uses *position embedding* for

every token. However, in PINMIR, we reset the position embedding of decoder for every intent description. In other words, the position at the start of every new intent description is equal for all intents. In that case, the decoder representations for every permutation of a given set of intents would be identical.

4.5 Experiments

We *extrinsically* evaluate NMIR on the **query facet generation** task. The task is defined as generating a number of textual facet descriptions for a given query. Following previous work on facet generation [42, 78], we focus on multi-faceted queries.

4.5.1 Evaluation Data

To evaluate this task, we use the MIMICS-Manual dataset [216]. This public dataset consists of 2464 unique web search queries sampled from the Bing query logs. The dataset contains between two and five facets for each query. The quality of each set of facets was manually assessed by three trained annotators. The quality labels are either Bad, Fair, or Good. In our experiments, we left out the Bad facet sets and considered the ones with either Fair or Good labels as our ground truth. Note that according to Zamani et al. [216], the Fair label still meets the quality criteria for being presented in a commercial web search engine. Although we find this a high-quality test collection for evaluating the performance of our model, we still present a small follow-up experiment with manual annotation to highlight the improvements compared to the baselines with a higher confidence.

Note that we have made sure that the intersection between the training and the test queries is empty. Similar to training, the top retrieved documents for each query in the test set was obtained from the Bing's Web Search API. For more information, see the training data details and training setup in Section 4.3.2.

Table 4.1: Results for the query facet generation experiment. All the improvements observed by NMIR compared to all the baselines are statistically significant.

#		Term Overlap			Exact Match			Set BLEU				Set BERT-Score		
facets	**Model**	**Prec**	**Recall**	**F1**	**Prec**	**Recall**	**F1**	**1-gram**	**2-gram**	**3-gram**	**4-gram**	**Prec**	**Recall**	**F1**
2	QDist	0.1637	0.1888	0.1676	0.0048	0.0046	0.0050	0.3841	0.1648	0.0438	0.0158	0.7649	0.7938	0.7807
	QFI	0.1936	0.2202	0.2033	0.0068	0.0061	0.0062	0.4070	0.1692	0.0515	0.0178	0.8057	0.8057	0.8007
	QFJ	0.2111	0.2023	0.2029	0.0072	0.0077	0.0072	0.4192	0.1835	0.0478	0.0076	0.8115	0.8020	0.8011
	QDMiner	0.2546	0.2369	0.2468	0.0089	0.0088	0.0088	0.5091	0.1931	0.0538	0.0089	0.8216	0.8162	0.8109
	BART	0.4621	0.5018	0.4888	0.0512	0.0500	0.0508	0.6413	0.6063	0.5709	0.5381	0.8616	0.8528	0.8540
	NMIR	**0.5195**	**0.6068**	**0.5539**	**0.1025**	**0.1040**	**0.1031**	**0.7333**	**0.6762**	**0.6403**	**0.6050**	**0.9170**	**0.9071**	**0.9062**
3	QDist	0.0929	0.1157	0.0957	0.0049	0.0045	0.0043	0.3518	0.1447	0.0341	0.0065	0.7418	0.7862	0.7366
	QFI	0.1330	0.1361	0.1337	0.0054	0.0052	0.0051	0.3868	0.1637	0.0407	0.0167	0.7916	0.8004	0.7797
	QFJ	0.1604	0.1801	0.1678	0.0065	0.0061	0.0064	0.3844	0.1695	0.0459	0.0135	0.7853	0.8021	0.7798
	QDMiner	0.1676	0.2024	0.2022	0.0082	0.0100	0.0084	0.4371	0.2014	0.0510	0.0169	0.7899	0.8100	0.7870
	BART	0.3672	0.4650	0.4193	0.0436	0.0410	0.0414	0.6025	0.5531	0.5040	0.4621	0.8390	0.8311	0.8293
	NMIR	**0.4279**	**0.5327**	**0.4687**	**0.0739**	**0.0720**	**0.0720**	**0.6960**	**0.6336**	**0.5949**	**0.5593**	**0.8840**	**0.8976**	**0.8775**
4	QDist	0.1725	0.2437	0.1876	0.0047	0.0044	0.0046	0.3843	0.1710	0.0543	0.0214	0.7674	0.7688	0.7769
	QFI	0.1951	0.2638	0.2223	0.0068	0.0064	0.0065	0.4014	0.1874	0.0642	0.0231	0.8005	0.8072	0.7969
	QFJ	0.1777	0.1454	0.1503	0.0064	0.0058	0.0060	0.3977	0.1800	0.0571	0.0212	0.7925	0.8047	0.7897
	QDMiner	0.1894	0.1672	0.1987	0.0065	0.0073	0.0068	0.4862	0.2230	0.0633	0.0230	0.8044	0.8040	0.7991
	BART	0.3165	0.4515	0.3896	0.0343	0.0348	0.0345	0.5940	0.5376	0.4611	0.4159	0.8222	0.8206	0.8175
	NMIR	**0.3898**	**0.5072**	**0.4358**	**0.0685**	**0.0677**	**0.0681**	**0.6940**	**0.6292**	**0.5899**	**0.5543**	**0.8802**	**0.8978**	**0.8775**
5	QDist	0.1557	0.1593	0.1440	0.0023	0.0024	0.0023	0.3387	0.1048	0.0439	0.0176	0.7165	0.7802	0.7192
	QFI	0.1605	0.1941	0.1720	0.0058	0.0050	0.0050	0.3539	0.1524	0.0523	0.0203	0.7603	0.8127	0.7584
	QFJ	0.1767	0.1348	0.1451	0.0055	0.0057	0.0053	0.3735	0.1675	0.0564	0.0234	0.7731	0.8136	0.7714
	QDMiner	0.2176	0.1443	0.1773	0.0069	0.0066	0.0065	0.4275	0.1826	0.0657	0.0234	0.7758	0.8036	0.7792
	BART	0.3043	0.4124	0.3558	0.0282	0.0263	0.0275	0.5087	0.4406	0.3969	0.3445	0.7633	0.8017	0.7660
	NMIR	**0.3877**	**0.4559**	**0.4121**	**0.0613**	**0.0584**	**0.0596**	**0.6313**	**0.5628**	**0.5222**	**0.4871**	**0.8442**	**0.8870**	**0.8405**
variable	QDist	0.0969	0.1564	0.1195	0.0017	0.0023	0.0019	0.1999	0.1134	0.0360	0.0107	0.6772	0.6855	0.6100
	QFI	0.1461	0.1748	0.1571	0.0057	0.0061	0.0059	0.2763	0.1269	0.0421	0.0140	0.7069	0.7113	0.6144
	QFJ	0.1807	0.2041	0.1894	0.0069	0.0067	0.0067	0.2484	0.1065	0.0242	0.0090	0.7196	0.6708	0.5871
	QDMiner	0.2060	0.2456	0.1894	0.0076	0.0083	0.0079	0.2893	0.1226	0.0301	0.0126	0.7220	0.7025	0.6285
	BART	0.4307	0.4618	0.4481	0.0474	0.0516	0.0486	0.4459	0.4003	0.3896	0.3351	0.7623	0.6932	0.6558
	NMIR	**0.4851**	**0.5673**	**0.4968**	**0.0790**	**0.0842**	**0.0784**	**0.5187**	**0.4748**	**0.4470**	**0.4192**	**0.8003**	**0.7487**	**0.6928**

4.5.2 Evaluation Metrics

To evaluate query facet generation models, we adopt four sets of evaluation metrics. (1) Term overlap metrics: these metrics have been previously used for evaluating query facet extraction models [76]. They include Term Precision (TP), Term Recall (TR), and Term F1-measure (TF). These metrics compute the precision, recall, and F1-measure for the set of terms generated by the model with respect to the terms appeared in the ground truth data. For more information about these metrics, refer to Kong and Allan [76]. (2) Exact match metrics: similar to term overlap, this metric also focuses on exact text matching but at the facet level. In other words, these metrics compute the precision, recall, and F1-measure of generating the exact facet description appeared in the ground truth. (3) Set BLEU scores: BLEU [127] is a widely adopted metric for text generation tasks, e.g., machine translation. However,

Table 4.2: Manual annotation results for pairwise comparison of NMIR vs. BART in facet generation.

Win	**Tie**	**Loss**
48%	30%	22%

it is defined between a single candidate text and a set of references. In our task, we deal with comparing two sets of text, one set is different facet descriptions generated by the model (R) and the other one is different facet descriptions in the ground truth test set (G). To compute Set BLEU, we first generate all permutations of R and then choose R^* such that $R^* = \arg\max_{R' \in \text{perm}(R)} \frac{1}{M} \sum_{i=1}^{M} \text{BLEU-4}(R'_i, G_i)$, where the subscript i denotes the facet index and $M = \max(|G|, |R|)$. We then compute the Set BLEU scores using $\frac{1}{M} \sum_{i=1}^{M} \text{BLEU-n}(R^*_i, G_i)$ for different n-grams. (4) Set BERT-Score: BERT-Score [222] has been recently used to compute the semantic similarity of a candidate text and a set of reference texts using the BERT representations [39]. We define Set Bert-Score as $\frac{1}{M} \sum_{i=1}^{M} \text{BERT-Score}(R^*_i, G_i)$. We compute this mean performance for all precision, recall, and F1-measures computed by the BERT-Score model.

4.5.3 Results and Discussion

We use the following baseline methods in our experiment:

- QDist [197]: QDist is a retrieval model that first generates multiple query variations and reformulations of the submitted query and learns a distribution over queries for retrieval. Even though this approach is not implemented for facet generation, its query variations can be seen as different query intents and can be used as a baseline for our model.

- QFI and QFJ [78]: We use the state-of-the-art variation of the QFI and QFJ methods [78] that were developed for facet extraction in web search. They are

Table 4.3: Some successful and unsuccessful examples of the facets generated by NMIR. Facets are separated using the ■ symbol.

Query	**Generated facets**
atropine sulfate drops	atropine sulfate drops interactions ■ atropine sulfate drops overdose ■ atropine sulfate drops precautions ■ atropine sulfate drops side effects ■ atropine sulfate drops uses
best fps games	best fps games for steam ■ best fps games for pc ■ best fps games for ps4 ■ best fps games for xbox one ■ best fps games for android
accident investigation	motorcycle accident ■ car accident ■ train accident ■ boat accident ■ forklift accident
awning window	awning window windows 10 ■ awned window windows 7 ■ a roofing window window windows 8 ■ a window windows vista ■ windows xp
balance of payment	balance of payment bank of america ■ balance of payments bank of ireland bank of germany bank of usa ■ bank of europe bank of philippines ■ cash balance

based on graphical models that estimate the probability of a hidden variable for modeling the extraction probability of each facet term. We followed the implementation details provided by the authors and selected the parameters using the validation set described in Section 4.3.2.

- QDMiner [42]: This is a competitive baseline for facet extraction from text and html documents. It is a hybrid approach that integrates multiple solutions for query facet extraction.

- BART [89]: We fine-tuned BART based on our training data, where the query and the top retrieved documents are the BART inputs and a concatenation of all query facet descriptions separated using a special token are the BART target output for training. Sequence-to-sequence models, like BART, provide strong performance for reformulation and facet generation tasks [119].

We emphasize that the QFI and QFJ models are shown to outperform other existing query facet extraction models [78]. There exist many methods that use metadata

or taxonomies to produce query facets, which are out of the scope of this chapter. For all the baselines, we follow the same hyper-parameter selection approach as the proposed model. Note that the main goal of this experiment is to provide extrinsic evaluation for the quality of the learned query intent representations. Therefore, we do not intend to show that NMIR is the state-of-the-art approach for facet generation, instead the goal is to demonstrate the quality of the learned representations through facet generation tasks.

The results are presented in Table 4.1. First, we observe that the proposed model consistently outperforms both probabilistic and neural baselines. This is true for all the evaluation metrics used in our experiment, including term matching, facet matching, n-gram matching, and semantic matching metrics. Note that all the improvements are statistically significant, according to the paired t-test with Bonferroni correction at 95% confidence.

We note that the test set for different number of facets is different. In other words, the numbers in different parts of Table 4.1 separated by a solid line should not be compared as their test queries are different. That being said, we still observe a consistent drop in performance as the number of facets increases, which makes sense considering the fact that it becomes increasingly more difficult.

Another observation is the large performance gap between QDist, QFI, QFJ, and QDMiner with the neural models (BART and NMIR). The reason is that the former are extractive facet generation models, while the latter are abstractive generation models. The ground truth contains several terms for describing the facets that are not in the result list, thus the extractive models fall short in generating them. This explains the poor performance of the extractive models.

The next observation from the result table is that the Exact Match performances are substantially lower than the other metrics. Exact Match is an extremely strict metric that only focuses on generating the exact facet description text used in the

ground truth. Term Overlap and Set BLEU provide smoother versions of term and phrase matching measures.

Furthermore, the results obtained by NMIR show that it achieves higher Term Overlap Recall than Precision, and this is consistent across all the test sets. This shows that the percentage of generated terms not included in the ground truth is larger than those in the ground truth missed by the model. Moreover, we observe that the performance of non-parametric NMIR for the variable facet number case is closer to its performance when the number of generated facets is equal to 2. The main reason is that the number of queries with 2-3 facets are dominated in the MIMICS-Manual dataset.

We further extend our evaluation using manual annotation. We showed a query to the annotators and asked them to review multiple pages of the result list for each query using a web search engine to understand different aspects of each query. We then showed them the facet descriptions generated by BART (our strongest baseline) and NMIR for the query and asked them to decide which one is a better facet description set, with respect to both quality and coverage. They could select one of them or vote for a tie. The presentation order (BART vs. NMIR) was random to reduce biases. We repeat this process for 100 queries randomly sampled from the test set by two annotators. In case of disagreement, we asked them to discuss and come up with an agreement or discard the query. The results for NMIR vs. BART are presented in Table 4.2. NMIR wins in 48% of the cases and loses in 22% of queries.

To study how permutation invariancy improves the effectiveness of NMIR is important to know the characteristics of our dataset. Each query in MIMIMCS contains between two and five facets. Most queries in this dataset only have two facets. Each query in our dataset contains an average of 2.81 facets per query. The results for our first set of experiments on this dataset are reported in Table 4.4 (# facets = variable). PINMIR generally outperforms all the baselines. The improvements in terms of exact

Table 4.4: Results for the query facet generation experiment. The superscript * denotes statistically significant improvements compared to all the baselines using two-tailed paired t-test with Bonferroni correction at 99% confidence level.

# facets	Model	Term Overlap			Exact Match			Set BLEU				Set BERT-Score		
		Prec	Recall	F1	Prec	Recall	F1	1-gram	2-gram	3-gram	4-gram	Prec	Recall	F1
variable	QDist	0.0969	0.1564	0.1195	0.0017	0.0023	0.0019	0.1999	0.1134	0.0360	0.0107	0.6772	0.6855	0.6100
	QFI	0.1461	0.1748	0.1571	0.0057	0.0061	0.0059	0.2763	0.1269	0.0421	0.0140	0.7069	0.7113	0.6144
	QFJ	0.1807	0.2041	0.1894	0.0069	0.0067	0.0067	0.2484	0.1065	0.0242	0.0090	0.7196	0.6708	0.5871
	QDMiner	0.2060	0.2456	0.1894	0.0076	0.0083	0.0079	0.2893	0.1226	0.0301	0.0126	0.7220	0.7025	0.6285
	BART	0.4307	0.4618	0.4481	0.0474	0.0516	0.0486	0.4459	0.4003	0.3896	0.3351	0.7623	0.6932	0.6558
	NMIR	0.4851	0.5673	0.4968	0.0790	0.0842	0.0784	**0.5187**	0.4748	0.4470	0.4192	0.8003	0.7487	0.6928
	PINMIR	**0.4891**	**0.5691**	**0.5107***	**0.0798**	**0.0856**	**0.0795**	0.5173	**0.4763**	**0.4491**	**0.4246***	**0.8173***	**0.7524***	**0.7199***
max	QDist	0.1557	0.1593	0.1440	0.0023	0.0024	0.0023	0.3387	0.1048	0.0439	0.0176	0.7165	0.7802	0.7192
	QFI	0.1605	0.1941	0.1720	0.0058	0.0050	0.0050	0.3539	0.1524	0.0523	0.0203	0.7603	0.8127	0.7584
	QFJ	0.1767	0.1348	0.1451	0.0055	0.0057	0.0053	0.3735	0.1675	0.0564	0.0234	0.7731	0.8136	0.7714
	QDMiner	0.2176	0.1443	0.1773	0.0069	0.0066	0.0065	0.4275	0.1826	0.0657	0.0234	0.7758	0.8036	0.7792
	BART	0.3043	0.4124	0.3558	0.0282	0.0263	0.0275	0.5087	0.4406	0.3969	0.3445	0.7633	0.8017	0.7660
	NMIR	0.3877	**0.4559**	0.4121	0.0613	0.0584	0.0596	0.6313	0.5628	0.5222	0.4871	0.8442	0.8870	0.8405
	PINMIR	**0.4712***	0.4302	**0.4423***	**0.0731***	**0.0689***	**0.0677***	**0.6505***	**0.5732***	**0.5411***	**0.4895***	**0.8731***	**0.8873**	**0.8740***

match are marginal, while we observe significant improvements for term overlap F1, BLEU 4-gram, and Set BERT-Score.

Intuitively, we expect a permutation-invariant loss to have higher impact on queries with more facets. In our second set of experiments, we solely focus on the queries with 5 facets (i.e., the maximum number of facets in MIMICS). According to Table 4.4, we observe substantially larger improvements in queries with five facets. The improvements are statistically significant in nearly all cases, except for term overlap recall and Set BERT-Score recall. This observation demonstrates that the permutation-invariant model has higher impacts on the queries with more intents.

We also propose the Stochastic Hungarian loss for efficiency reasons. In our experiments, we observe no statistically significant difference between the effectiveness of a model trained with Hungarian loss compared to its stochastic variation (with three samples). Hungarian loss achieves a term overlap F1 of 0.4724 for queries with three facets while this value for the Stochastic Hungarian loss is 0.4731. We made similar observations for other metrics. Therefore, both exact and stochastic Hungarian losses perform comparably, but the stochastic variation can be used for larger number of facets efficiently.

4.6 Summary

In this chapter, we introduced NMIR, a general retrieval-augmented framework that uses the top retrieved documents for learning multiple representations for each input query. These multiple representations are used to better represent faceted and ambiguous queries. We implemented the proposed framework using the state-of-the-art encoder-decoder architectures, e.g., BART, for initializing the encoder and decoder parameters and Guided Transformer for mapping a generic query representation to an intent representation space. We also introduced an asynchronous optimization approach for efficient training of the framework. Our evaluation on the query facet generation task demonstrated the effectiveness of the proposed solution compared to competitive baselines.

Despite its strong performance, NMIR suffers from some design limitations. In particular, the NMIR's solution for achieving multiple representations for a query is to generate all the query intents associated with the query. However, the model expects the output to be exactly in the same order as it appears in the ground truth. We further addressed this issue by proposing a permutation-invariant variant of the NMIR framework, named PINMIR. This model learns to generate a set of text pieces in a permutation-invariant manner. To this aim, we introduced a stochastic Hungarian loss function for learning multiple permutation-invariant query representations. By resetting the positional embedding for each intent description generated by the model, PINMIR ensures that the decoder is also permutation-invariant. We showed that this approach leads to further improvements.

CHAPTER 5

ADAPTING RETRIEVAL MODELS USING TARGET DOMAIN DESCRIPTION

In this chapter, we introduce a new domain adaptation category for information retrieval – the task of domain adaptation using the target domain description. In the following, we first motivate the task (Section 5.1), and then we define a taxonomy of domain attributes in retrieval tasks to understand different properties of a source domain that can be adapted to a target domain. Our experiments show that a retrieval-augmented approach for domain attribute-value extraction based on the defined taxonomy can effectively identify various properties of each target domain, including the topic of documents, their linguistic attributes, and their source. We propose a novel automatic data construction pipeline that produces a synthetic document collection, query set, and pseudo relevance labels, given a textual domain description.

5.1 Motivation

The effectiveness of neural information retrieval models has been well-established in recent years [25, 46, 112]. However, these models have primarily demonstrated strong performance in settings where the training and test data follow a similar data distribution [171]. When well-performing neural models developed for one test collection, e.g., MS MARCO [17], are applied to a substantially different one, the results

---

The content of this chapter is largely based on our article published in the proceedings of ACM ICTIR 2023 [54].

are often worse than those produced by much simpler bag-of-words models such as BM25 [148]. This poses a problem in real-world applications, where access to large, domain-specific training data is limited. For a general description of this problem in machine learning, refer to Hand [48]. To address this issue, a group of methods known as "domain adaptation" have been developed.

There are various approaches to domain adaptation in information retrieval, as summarized in Table 5.1. In the zero-shot setting, the assumption is that the model has been trained on a large-scale test collection in a source domain, but no data from the target domain is available during training. It is worth noting that in the zero-shot setting, there is no adaptation taking place, as the model is simply being tested on the target domain. In contrast, unsupervised domain adaptation models assume that the target document collection is available for adaptation. The few-shot setting takes this further and assumes that a small set of query-document pairs with relevance labels on the target domain is available, allowing the retrieval model to be adapted to the target.

In this chapter, we introduce a new category of domain adaptation methods for neural information retrieval, which we refer to as "domain adaptation with description." Studying this problem is not only interesting from an academic perspective, but also has potential applications in real-world scenarios, where the target collection and its relevance labels are not available at training time. For example, these may not be available yet or at all or, even if they were, target domain owners may be hesitant to provide them for several reasons, such as legal restrictions. There are also applications with privacy concerns, for instance in the case of medical records or where the data contains personally identifiable information. Another example can be found when a competitive advantage is involved, as potential use of the data may benefit competitors. Therefore, if an organization lacks the resources for training neural IR models in-house and desires to outsource the process, they should be able to provide a

Table 5.1: Different categories of domain adaptation in information retrieval.

Adaptive Retrieval Setting	q-d-$r^{\dagger}$ **in** D_1	q-d-r **in** D_2	**Target Corpus**	**Extra Information**
Zero-shot retrieval	✔	✗	✗	None
Unsupervised domain adaptation	✔	✗	✔	None
Supervised domain adaptation	✔	✔‡	✔	None
Domain adaptation with description	✔	✗	✗	textual description of the target domain*

$\dagger$ $q-d-r$ refers to training data triplets of query, document, and relevance labels.
$\ddagger$ often only a small amount of training data is available.
$*$ domain description can be a single sentence describing the target domain.

high-level textual description that outlines the task and characteristics of the data in a general manner. Our approach then allows the organization to convey the necessary information to a third party without compromising sensitive information or violating legal restrictions.

In this chapter, we investigate the task of domain adaptation for information retrieval by utilizing target domain descriptions. We propose a taxonomy for the task and analyze the attributes by which a domain can be adapted. We differentiate our task from related recent studies and explain the limitations of existing technologies. To address these limitations, we propose a novel pipeline that utilizes the domain descriptions to construct a synthetic target collection and generate queries and pseudo relevance labels to adapt the initial ranking model trained on a source domain. Our approach takes advantage of state-of-the-art instruction-based language models to extract the properties of the target domain based on its given textual description. We show that a retrieval-augmented approach for domain attribute-value extraction can effectively identify various properties of each target domain, including the topic of documents, their linguistic attributes, and their source. The extracted properties are used to generate a seed document using generative language models and then an iterative retrieval process is employed to construct a synthetic target collection, automatically.

Following prior work on unsupervised domain adaptation [185], we automatically generate queries from our synthetic collection based on the query properties extracted from the target domain description. We then generate pseudo-relevance labels for each query given an existing cross-encoder re-ranking model and use the created data

for adapting *dense retrieval* models to the target domain. Extensive experiments on five diverse target collections, ranging from financial question answering to argument retrieval for online debate forums, demonstrate the effectiveness of the proposed approach for the task of domain adaptation with description. In summary, the main contributions of this work include the following.

- Introducing the novel task of domain adaptation with description for information retrieval.
- Proposing an automatic data construction pipeline from each target domain description.
- Proposing a taxonomy of domain attributes in information retrieval for developing effective domain adaptation methods.
- Studying a retrieval-augmented approach based on state-of-the-art language models for extracting the attributes in our taxonomy from domain descriptions.
- Introducing an effective implementation of the proposed pipeline.
- Significantly outperforming competitive applicable baselines on five diverse retrieval benchmarks.

5.2 Methodology

In this section, we explain the problem formulation and a taxonomy of domain attributes that can be used to understand domain descriptions. Such domain attribute-value extraction component can produce attribute values for a synthetic corpus construction model that uses a large language model to generate one seed document with these attributes and then performs an iterative retrieval process from a heterogeneous collection such as the Web for collection creation. The constructed collection will then be used to generate queries and pseudo relevance labels that are aligned

Table 5.2: A taxonomy of attributes that define an information retrieval task.

	Retrieval Attribute	Attribute Definition	Example Attribute Values
Query Attributes	Query topics*	the subject matters or themes of the users' search requests	medical, financial, climate, etc.
	Query linguistic features	syntactic characteristics of the query	formal, informal, technical, etc.
	Query language	a language used by the user to make requests for information	English, Spanish, etc.
	Query structure	the structure of the query used by the user	structured, semi-structured, unstructured, SQL, etc.
	Query modality	the query modality	text, text and image, uni-modal, multi-modal, etc.
	Query format	type of the query submitted by the user	keyword queries, tail queries, tip-of-tongue queries, etc.
	Query context	any metadata that exists around the query	conversational search, session search, from adult users vs. kids
Doc Attributes	Document topics	the main subjects that the document collection cover	medical, financial, etc.
	Document linguistic features	syntactic characteristics of the documents	formal, informal, technical, etc.
	Document language	the language used to express the content of the documents	English, Spanish, etc.
	Document structure	the structure of the documents in the collection	structured, semi-structured, unstructured, knowledge base, etc.
	Document modality	the document modality	text, text and image, uni-modal, multi-modal, etc.
	Document format	the format of the document (especially from IR perspective)	passages, long documents, questions, etc.
	Document source	the specific source that the documents come from	Wikipedia, Twitter, Quora, etc.
	Relevance notion	the criteria that make the documents relevant to the query	topical relevance, containing the correct answer, paraphrasing, containing the counterargument, etc.

* This is often referred to as the "domain", but we use the term "topic" to avoid confusion.

with the properties of the target domain, as extracted by our domain attribute-value extraction component. This pipeline leads to a synthetic training set that can be used to adapt a dense retrieval model to the target domain.

5.2.1 Problem Formalization

Let M be a retrieval model that is trained on the source domain D_1, and T be the textual description of the target retrieval domain D_2, where $D_2 \neq D_1$. The goal is to adapt the retrieval model M to the target domain D_2 and obtain the retrieval model M' that performs effectively on D_2. Assume that W is a large-scale heterogeneous collection, such as a Web collection, which can be used as an external resource. This large-scale collection can then be used for synthetic collection construction for any target domain description.

5.2.2 A Taxonomy of Domain Attributes in IR

The term "domain" is used quite loosely in NLP and IR and defined in myriad ways [135]. It often describes a type of corpus that is "coherent," such as a specific topic or linguistic register [134]. However, the concept of domain has evolved in recent years, leading to ongoing research in this area. For example, there is a distinction between "canonical" data (e.g., edited news articles) and "non-canonical" data (e.g., social media), and models trained on one type may not perform well on the other. There is an ongoing debate over what constitutes a "domain" in the field of information retrieval (IR), and whether subdomains exist within a larger domain. This uncertainty makes it difficult to tackle the domain adaptation problem and develop a universal algorithm, as domain shifts are specific to each case and models may not perform robustly when transferred from one case to another.

To clarify the different stances on the definition of a "domain" we have developed a taxonomy for domains and their attributes in the context of IR. Therefore, we define a domain based on the set of attributes defined in our taxonomy. This taxonomy can be used to develop general-purpose domain adaptation solutions as it enumerates the possible ways in which two domains can be different. We argue that every retrieval task is composed of three variables: query, documents, and relevance notion. We propose that attributes related to these three categories together define a retrieval domain. In other words, for any domain D, we define a set of attributes $\{a_1, a_2, \cdots, a_n\}$, where each attribute a_i is either related to the properties of query, document, or relevance. Through careful exploration of many different retrieval tasks, including the ones in the BEIR benchmark [171] and the ones organized by TREC and CLEF evaluation campaigns over the last few decades, we compile a taxonomy that includes seven query-level attributes, seven document-level attributes, and one attribute denoting the relevance notion. The attributes, their definition, and examples are presented in Table 5.2. We argue that if the value of at least one at-

tribute belonging to any of the three categories changes, a domain shift has occurred. We highlight the asymmetric nature of query and document attributes that presents unique challenges for domain adaptation in IR compared to NLP tasks. Finally, we note this taxonomy can be used to see what attributes differ between domains and that we can leverage those for effective adaptation.

5.2.3 Domain Attribute-Value Extraction

As discussed earlier in this chapter, clients may be reluctant to provide actual target domain data. However, providing a high-level description of the data is usually feasible. In our problem, we need a description of the *retrieval task* that includes information on the appearance of the corpus and queries, in addition to user intentions, and how relevance is defined for that task. To obtain these descriptions, we gave 15 diverse IR collections from the BEIR benchmark [171] to three IR experts and asked them to explain the retrieval task for each. We asked them to revise the differences of opinion during a brainstorming session; they shared their explanations and worked together to reach a single description for each collection, which we refer to as T in our formalization. After the descriptions are finalized, we provide the same people with the taxonomy we have defined in Table 5.2, and ask them to annotate the descriptions based on the taxonomy attribute. This annotation results in the gold labels of attribute values based on our taxonomy for each dataset. We provide one dataset description and its annotation in Table 5.3 for the reference.

We argue that a proper understanding of the description has a significant impact on adaptation. If the model extracts the value of each attribute in the taxonomy, it knows when a domain shift has occurred and what attributes need to be adapted for the entire model to be adapted. Therefore, our domain attribute-value extraction component focuses on predicting the values of attributes defined in our taxonomy. Since the value of the attributes can be open-ended text rather than defined options,

Table 5.3: An example of a retrieval task description and its annotated attribute-value pairs from our taxonomy.

Target Collection	Arguana
Description of the retrieval task	Given an argument passage as a query, the task is to retrieve passages from online debate portals that contain its counterarguments
Description annotation	relevance notion: counterargument ■ query topic: NA ■ query linguistic features: NA ■ query language: NA ■ query structure: unstructured ■ query modality: unimodal ■ query format: argument passage ■ document topic: NA ■ document linguistic features: NA ■ document language: NA ■ document structure: unstructured ■ document modality: unimodal ■ document format: argument passage ■ document source: online debate portals

the best architectural choice is a text generation model that takes the domain description as input and generates the value of the attributes as output. Therefore, we adopt a state-of-the-art prompt-based text generation model F to perform the task, i.e., ChatGPT. We instruct the model to get the description of the domain and extracts the value of attributes introduced in the taxonomy.[1]

In addition to the instruction, we include up to three examples from the most similar collections to the target domain by retrieval augmentation. Let $R(T, C')$ denote a retrieval model (SBERT in our case) that takes the target domain description and a collection of textual descriptions of different domains (C'). The domain attribute-value extraction function F takes the instruction I, the retrieved examples, and domain description T, and outputs the values of attributes introduced in taxonomy. Formally: $F(I, T, R(T, C')) = \{a'_1, a'_2, \cdots, a'_n\}$ where $n = 15$.

[1]After some rounds of trial and error, we landed on the following instruction, I as the best performing one for our task: "For each defined retrieval task in the Passage, find the values related to the relevance notion (e.g., topically relevant, contains the answer, references of a paper, paraphrase, evidence for the claim, etc.) as well as the following query and document attributes: query topic (e.g., medical, scientific, financial, mathematical, adult, etc.); query linguistic features (e.g., formal, informal, etc.); query language (e.g., english, french, etc.); query structure (e.g., unstructured, semi-structured, structured, etc.); query modality (e.g., text, image, video, etc.); query format (e.g., keyword query, tail query, question, claim, argument, passage, etc.); document topic (e.g., medical, scientific, financial, mathematical, adult, etc.); document linguistic features (e.g., formal, informal, etc.); document language (e.g., english, french, etc.); document structure (e.g., unstructured, semi-structured, structured, etc.); document modality (e.g., text, image, video, etc.); document format (e.g., passage, long document, question, etc.); document source (e.g., StackExchange, wikipedia, reddit, youtube, twitter, facebook, quora, etc.). If the value of each attribute cannot be inferred, return NA"

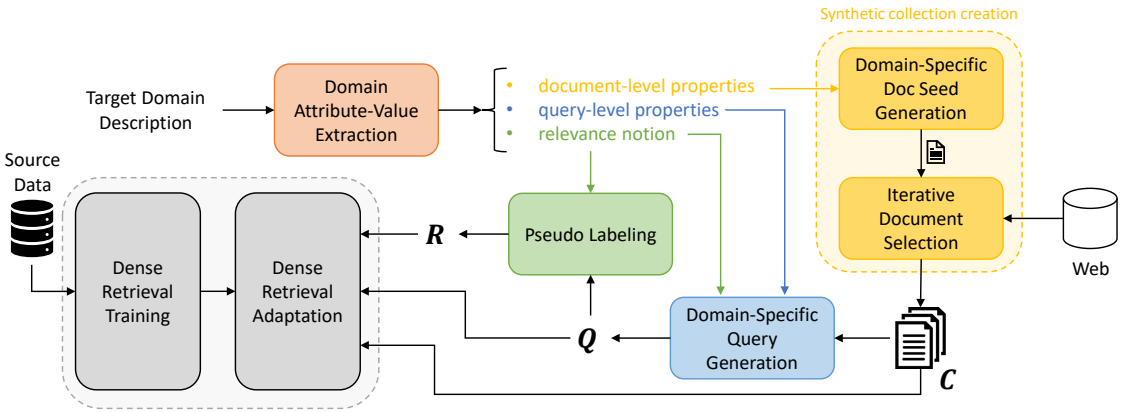

Figure 5.1: The proposed pipeline for training dense retrieval models for a given domain description.

Discussion One may argue that the taxonomy is easy to understand and interpret, therefore, users can directly identify these properties for the target domain and this bypasses the need to a domain attribute-value extraction component. This argument is valid. In other words, the taxonomy we define in Table 5.2 enables users of the system to directly identify the values of each attributes for the target domain. That being said, the domain attribute-value extraction component enables the users to just describe their target domain in natural language. Similar to any semantic parsing task, such as text-to-SQL, this component creates a natural language interface for this task. Thus, studying it can shed light into how feasible it is to extract domain attributes from natural language.

5.2.4 Synthetic Target Data Construction

As depicted in Figure 5.1, once we identify the domain attributes of our taxonomy for the target domain (i.e., domain attribute-value extraction), we propose to build a synthetic training set based on the generated attribute values. This consists of three steps: synthetic document collection construction, synthetic query generation,

and pseudo-labeling. In the following we describe each of these steps. Our data construction approach is presented in Algorithm 1.

Algorithm 1 Our Synthetic Data Creation Approach

Input (a) T: a target domain description; (b) W: a large, diverse, and heterogeneous collection (such as the Web); (c) M_θ: a dense retrieval model trained on the source domain; (d) $\widehat{M}$: an effective teacher model for pseudo labeling; (e) N: the desired size of synthetic collection ; (f) k: the iterative retrieval list size; (g) k': the number of generated queries per document.
Output A dense retrieval model M' for the target domain.
$a_1, a_2, \cdots, a_{15} \leftarrow$ AttributeValueExtraction(T)
$q_{\text{attr}} \leftarrow \{a_1, a_2, \cdots, a_7\}$
$d_{\text{attr}} \leftarrow \{a_8, a_9, \cdots, a_{14}\}$
$r_{\text{attr}} \leftarrow \{a_{15}\}$
$S_{\text{seed}} \leftarrow$ DocumentGen(d_{attr})
$C \leftarrow \emptyset$
repeat
    $d \leftarrow S_{\text{seed}}.pop()$
    $D \leftarrow$ Retrieve(query $= d$, collection $= W$, count $= k$)
    $C \leftarrow C \cup D$
    $S_{\text{seed}} \leftarrow S_{\text{seed}} \cup D$
until $|C| < N$
$Q \leftarrow$ QueryGen($C, q_{\text{attr}}, r_{\text{attr}}, k'$)
$R \leftarrow$ PseudoLabeling($C, Q, r_{\text{attr}}, \widehat{M}$)
$M' \leftarrow \arg\min_\theta \mathcal{L}(M_\theta, \{Q, C, R\})$
return M'

5.2.4.1 Synthetic Document Collection Construction

One naive approach to synthesizing the collection is to generate documents one by one using sequence-to-sequence models. In preliminary experiments, we observed that many state-of-the-art and free-to-use sequence-to-sequence models such as the latest version of Tk-Instruct [189], are not sufficient to generate meaningful documents given our target domain descriptions. Instead, they generate passages containing words from our instructions, rather than generating a document with the provided attributes.

It can be argued that with the rise of black-box generative language models like ChatGPT, this issue will be reduced. However, it is important to note that these models are not free to use. At the time of conducting this research, ChatGPT was not yet available through an API, so we used the next best available large language model, `text-davinci-003`, the latest version of GPT-3 from OpenAI. At the time of this research, OpenAI was charging customers based on the cumulative number of tokens in the input and output, at a rate of $0.02 per 1K tokens. If we consider an average passage to be 300 tokens, the minimum cost to generate a corpus like MS MARCO (consisting of 8M passages) would be $12,000. This assumes the model only takes the domain description with no example as input and generates one passage in line with the target domain description.

It is worth noting that our preliminary experiments showed the `text-davinci-003` model was unable to generate a desired passage even with three examples in the prompt. Additionally, these models cannot perform a sequence of tasks step by step (e.g., curating a collection then queries, etc.). They may miss some parts of the sequence or do it all at once (generating documents and queries simultaneously), causing the automation of the training retrieval model to be difficult.

To overcome all these obstacles, we propose an iterative document selection process (i.e., lines 7-14 in Algorithm 1). We first generate a document based on the domain attributes we extracted from the target domain description T. We call this generated document a seed document. We find that ChatGPT is the only language model that could successfully generate a related document given our document attributes. We tried T5, Tk-INSTRUCT, and GPT-3 and they could not generate a document with the given attributes. Instead, they generate a text using the words in the given instruction which is not sufficient for effective domain adaptation. We then run an iterative retrieval process using BM25 and a BERT-based cross-encoder reranking model trained on the source domain [121]. It retrieves k documents (we em-

pirically observe that k should be set to a small value often less than 50) in response to the seed document and then adds all the retrieved documents to the seed set. Again, another document from the seed set is selected and another k documents are retrieved. This process repeats until we reach a collection C with a desired synthetic collection size (N).

5.2.4.2 Synthetic Query Generation

In line 15 of Algorithm 1, we generate k' queries per document in the constructed document collection C. To this aim, we train instruction-based $T5$ on MS MARCO for query generation using the MS MARCO query and relevance attributes. It is similar to the docT5query [118], but also takes query and relevance properties of the target domain as input. To be precise, we use form this input for the instruction-based $T5$ model: 'Generate a query for the following Passage based on the given Attributes. Passage: $\cdots$. Attributes: $\cdots$.' We include the query and relevance attributes in the instruction. Therefore, it learns to generate queries with the given properties. The model is trained with a maximum likelihood objective as follows:

$$-\sum_{k} \log P(q_k|q_{i<k}, q_{\text{attr}}, r_{\text{attr}}),$$

where q_k is k^{th} output query token, q_{attr} is the extracted values for query attributes in the taxonomy, and r_{attr} is the extracted values for relevance attribute. We use beam search with the size of k'.

5.2.4.3 Pseudo Labeling

Research on weak supervision [38, 212] showed that we can use existing retrieval models to annotate documents for a given query set and train student models based on the annotated data. More recently, this approach has been found effective in unsupervised domain adaptation [185]. We use a cross-encoder re-ranking model

based on BERT [121] that is trained on MS MARCO (our source domain) as a teacher model and annotate documents through soft labeling: the input includes the query, the relevance notion, and a document, and the output scores are used as labels. Let $D_q \subset C$ be a set of documents that should be annotated for query q by the pseudo-labeler. We construct D_q as follows:

- D_q includes the document that q was generated from.
- D_q includes 25 random documents from the top 100 documents retrieved by BM25.[2]
- D_q includes 25 random documents from the top 100 documents retrieved by the dense retriever M_θ.

5.2.5 Dense Retrieval Adaptation

Given the constructed training set with pseudo-labels, we use the following listwise loss function for adapting the dense retrieval model M_θ to the target domain. We used Contriever [62] (an unsupervised dense retrieval model trained using contrastive learning) that is fine-tuned on MS MARCO as our M_θ. Let $D_q \subset C$ be the set of documents annotated for query $q \in Q$ through pseudo-labeling. We use the following listwise loss function for each query q:

$$\sum_{d,d' \in D_q} \mathbb{1}\{y_q^T(d) > y_q^T(d')\} \left| \frac{1}{\pi_q(d)} - \frac{1}{\pi_q(d')} \right| \log(1 + e^{y_q^S(d') - y_q^S(d)}),$$

where $\pi_q(d)$ denotes the rank of document d in the result list produced by the student dense retrieval model, and $y_q^T(d)$ and $y_q^S(d)$ respectively denote the scores produced by the teacher and the student models for the pair of query q and document d. This knowledge distillation listwise loss function is inspired by LambdaRank [15] and is also used by Zeng et al. [218] for dense retrieval distillation.

[2]We empirically observe that taking 25 random samples from the top 100 documents leads to more robust performance compared to using the top 25 documents.

Table 5.4: Domain adaptation results in terms of NDCG@10 and Recall@100. Bold numbers indicate the highest value in each column (excluding Oracle). The superscript * denotes statistically significant improvements compared to all the baselines with respect to a two-tailed paired t-test with Bonferroni correction ($p\_value < 0.05$).

Model	TREC COVID		FiQA		SciFact		ArguAna		Quora	
	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100
BM25	0.688	**0.498**	0.253	0.539	0.690	0.908	0.471	0.942	0.807	0.973
ANCE	0.652	0.457	0.295	0.581	0.511	0.816	0.418	0.934	0.852	0.987
SBERT	0.477	0.072	0.257	0.542	0.537	0.846	0.425	0.945	0.855	0.988
Contriever	0.273	0.172	0.245	0.562	0.649	0.926	0.379	0.901	0.835	0.987
Contriever-FT	0.596	0.407	0.329	0.656	0.677	0.947	0.446	0.977	0.865	0.993
HyDE	0.593	0.414	0.273	0.621	0.691	**0.964**	0.466	**0.979**	-	-
ANCE - Cond. Query	0.640	0.459	0.294	0.575	0.518	0.813	0.406	0.932	0.843	0.980
Contriever-FT - Cond. Query	0.596	0.409	0.336	0.652	0.667	0.949	0.445	0.966	0.866	0.980
Ours	**0.737***	0.481	**0.344***	**0.684***	**0.695***	0.957	**0.497***	0.967	**0.881***	**0.995**
Oracle	0.752	0.515	0.368	0.699	0.744	0.970	0.529	0.973	0.885	0.984
CE Reranker	0.757	0.498	0.347	0.539	0.688	0.908	0.311	0.942	0.825	0.973

Table 5.5: Ablation Study in terms of NDCG@10 and Recall@100. Bold numbers indicate the highest value in each column (excluding Oracle). The superscript $^\blacktriangledown$ denotes statistically significant performance degrade compared to our method (the first row of the table). Significance is identified using a two-tailed pair t-test with Bonferroni correction ($p\_value < 0.05$).

Model	TREC COVID		FiQA		SciFact		ArguAna		Quora	
	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100	NDCG@10	R@100
Ours	**0.737**	**0.481**	**0.344**	**0.684**	**0.695**	**0.957**	**0.497**	**0.967**	**0.881**	**0.995**
Ours w/o pseudo-labeling	0.691$^\blacktriangledown$	0.473	0.336$^\blacktriangledown$	0.671$^\blacktriangledown$	0.687$^\blacktriangledown$	0.907$^\blacktriangledown$	0.477$^\blacktriangledown$	0.919$^\blacktriangledown$	0.852$^\blacktriangledown$	0.963$^\blacktriangledown$
Ours w/o seed document generation	0.688$^\blacktriangledown$	0.399$^\blacktriangledown$	0.310$^\blacktriangledown$	0.660$^\blacktriangledown$	0.630$^\blacktriangledown$	0.874$^\blacktriangledown$	0.441$^\blacktriangledown$	0.882$^\blacktriangledown$	0.822$^\blacktriangledown$	0.919$^\blacktriangledown$
Ours w/o *interactive* synthetic corpus creation	0.704$^\blacktriangledown$	0.478	0.343	0.638$^\blacktriangledown$	0.662$^\blacktriangledown$	0.935$^\blacktriangledown$	0.481$^\blacktriangledown$	0.954	0.841$^\blacktriangledown$	0.993

In addition, we take advantage of the other passages in the batch as in-batch negatives. Although in-batch negatives resemble randomly sampled negatives that can be distinguished easily from other documents, it is efficient since passage representations can be reused within the batch [67].

5.3 Experiments

This section describes our datasets, experimental setup, and results.

5.3.1 Tasks and Data

For evaluating our domain adaptation solution, we chose the target collections to be as diverse as possible within the public test collections in the BEIR benchmark [171]. Below we provide brief explanations of these collections.

Source Domain As the source domain, we focus on passage retrieval provided by the MS MARCO collection [17]. As the standard practice on zero-shot learning offered by BEIR benchmark, most baselines models have been pre-trained on this dataset, as our source domain. It contains 8.8 M passages and an official training set of 532,761 query-passage pairs collected from the Bing search log. Queries often have only one relevant passage per query with binary relevance label.

Target Retrieval Task 1: Bio-Medical IR Our first target retrieval task focuses on retrieving scientific documents for biomedical queries. We use the collection provided by the TREC Covid Track in 2020 (**TREC-COVID**) [181], which is an ad-hoc retrieval task based on scientific documents related to the Covid-19 pandemic offered by the CORD-19 corpus [186]. Similar to Thakur et al. [171], we use the July 16, 2020 version of CORD-19 collection as the target corpus, and the final cumulative judgments with query descriptions from the original task as test queries. The test collection consists of 50 test queries and a corpus of 171K documents.

Target Retrieval Task 2: Financial Question Answering Our second task studies answer passage retrieval in response to natural language questions in the financial domain. We use the FiQA-2018 Task 2 [106] (**FiQA**) that focused on answering questions based on personal opinions. The document collection was created by crawling posts on StackExchange under the Investment topic from 2009-2017, which serves as the corpus with 57K documents. The test set consists of 648 queries.

Target Retrieval Task 3: Argument Retrieval This task explores ranking argumentative texts from a collection based on relevance to a given query on various subjects. We use the **ArguAna** dataset [182] which has passage-level queries. The goal is to retrieve the most suitable counterargument for a given argument. The collection was collected from online debate portals. There are 1,406 argument queries in the dataset and the corpus size is 8.67K.

Target Retrieval Task 4: Duplicate Question Retrieval : The aim of duplicate question retrieval is to detect repeated questions asked on community question-answering (CQA) forums. We use the **Quora** dataset that consists of 522,931 unique questions in corpus and 10,000 test queries.

Target Retrieval Task 4: Fact Checking Fact checking involves verifying a statement against a large pool of evidence. It requires knowledge of the statement and the ability to analyze multiple documents. In a retrieval setting, the query is a claim, and we attempt to retrieve documents that confirm or refute the claim. We use the **SciFact** collection [183] that consists of 300 scientific claims as test queries and 5K paper abstracts as the corpus.

Constructing the heterogeneous Collection W: As explained in Section 5.2.1, W is a heterogeneous collection of documents from which our model selects documents to synthesize the target retrieval corpus. To create this collection, we ensure that there is no document leakage between the target retrieval tasks and W.[3] We create W by putting together the documents from MS MARCO [17], SciDocs [22], NFCorpus [13], Touche-2020 [10], and CQADupStack [59]. This results in a collection with 9M+ documents.

5.3.2 Experimental Setup and Evaluation Metrics

We implemented and trained our models using TensorFlow. The network parameters were optimized using Adam [73] with linear scheduling and the warmup of 4,000 steps. The learning rate was selected from $[1 \times 10^{-6}, 1 \times 10^{-5}]$ with a step size of 1×10^{-6}. The batch size was set to 128. We set k to 30, N to 10,000, and k' to 5 (see

[3]Note that document leakage is not necessary an issue in this task. In the real world, the Web contains several types of documents that can satisfy the attributes of each target domain (e.g., each BEIR collection). The main challenge is to identify and recover these documents from a large heterogeneous corpus.

Algorithm 1). We use the BERT [39] with the pre-trained checkpoint made available from Contriever-FT [62] as the initialization. Hyper-parameter selection (for both BM25 and neural models) and early stopping was conducted based on the performance in terms of MRR on the MS MARCO validation set. For query generation we use the T5 model from Nogueira et al. [118]. As the re-ranking teacher model for pseudo labeling, we use a BERT cross-encoder [121]. For domain attribute-value extraction, we use three examples in the ChatGPT instruction. Following BEIR [171], we use NDCG@10 and Recall@100 as evaluation metrics. We use a two-tailed paired t-test for identifying statistically significant performance differences using Bonferroni correction with $p\_value < 0.05$.

5.3.3 Results and Discussion

We compare our method against the following baselines:

1. BM25 [148]: an effective term matching retrieval method that evaluates and ranks a group of documents based on the presence of query terms regardless of their position in each document.

2. ANCE [195]: a bi-encoder dense retrieval model that constructs hard negatives from an Approximate Nearest Neighbor (ANN) index of the corpus based on the model's representations. Consistent with previous works, we used RoBERTa [99] as the base language model that is trained on MS MARCO for 600K steps for our experiments.

3. SBERT [147]: another dense retrieval baselines that uses BERT that employs Siamese and Triplet network architectures to generate sentence embeddings.

4. Contriever [62]: an unsupervised dense retrieval model that learns adaptive representation via contrastive learning.

5. Contriever-FT [62]: the Contriever model that is fine-tuned on MS MARCO training set.

6. HyDE [44]: it utilizes GPT-3 to generate a hypothetical document. Then it uses Contriever to retrieve from the corpus with the hypothetical document as the query. This is a concurrent work to ours.

7. ANCE - Cond Query: following Asai et al. [3], which is another concurrent work to ours, we concatenate the domain description with the query in ANCE so the query encoder is aware of the domain description.

8. Contriever-FT - Cond Query: this is similar to the last baseline, but uses Contriever-FT as the dense retrieval model.

As a source of reference we compare against the following approaches:(1) Oracle: this is our proposed approach that, instead of document collection construction, uses the target domain collection for query generation; and (2) CE Reranker: this is a BERT-based cross-encoder reranker trained on MS MARCO, which reranks the top 100 documents returned by BM25. Since this is not a dense retrieval model, we only report its results as a point of reference.

The results are reported in Table 5.4. We observe that dense retrieval baselines have difficulties surpassing the BM25 performance on TREC COVID, SciFact, and ArguAna datasets in terms of NDCG@10 in a zero-shot setting. This demonstrates the difficulty of dealing with distribution shift in neural information retrieval. HyDE that uses GPT-3 for generating hypothetical documents for test queries performs well in terms of Recall@100 on SciFact and ArguAna datasets. The proposed approach outperforms all dense retrieval baselines in terms of NDCG@10 in all collections. These improvements are statistically significant in all cases. It is also better than its counterparts in terms of Recall@100 on FiQA and Quora. Interestingly, our approach

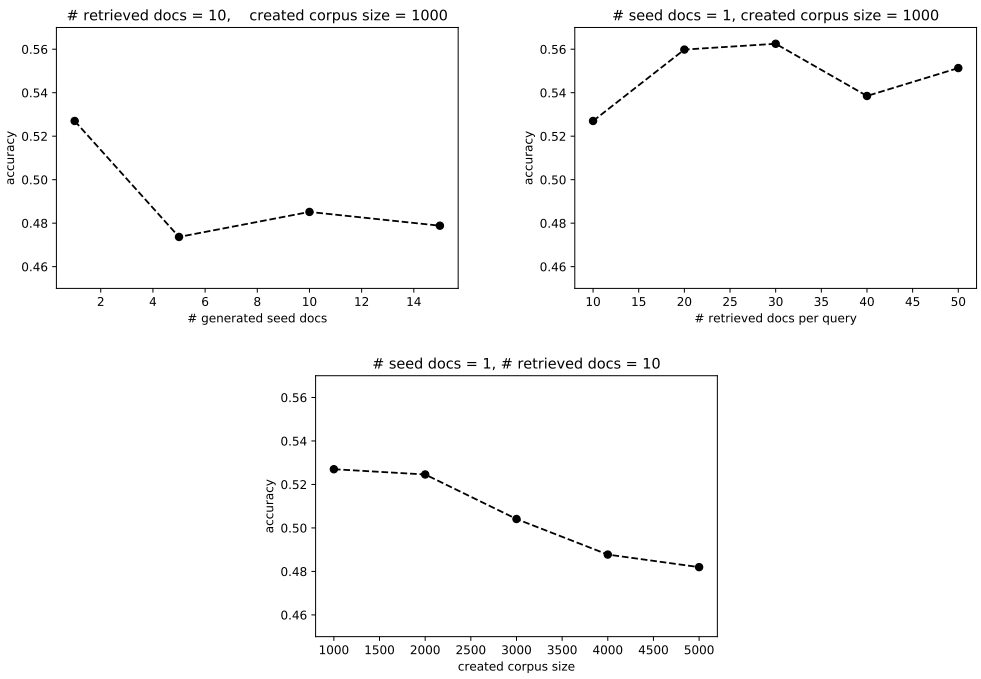

Figure 5.2: Sensitivity of our iterative corpus creation process to different parameters in terms of average accuracy.

is the only dense retrieval model that can beat BM25 on TREC COVID and ArguAna. This demonstrates the effectiveness of our data creation pipeline.

The performance gap between the Oracle model and the baselines is often less than 10%, confirming the quality of the synthetic corpus our model creates. The Oracle model performs better than the proposed approach in all cases, except for Recall@100 on Quora. Note that the Oracle model does not necessarily provide upper-bound results, it just uses the target domain collection instead of synthetic collection construction. These results suggest that it is likely to construct a collection that dense retrieval models benefit from for adaptation, even more than the actual target collection. Our model outperforms the cross encoder reranker model in terms of Recall@100 in all cases, except for TREC COVID.

Ablation Study. To demonstrate the impact of each design decision we made in our pipeline, we ablate each major component in our model and report the results in

Table 5.5. We first exclude the pseudo-labeling component (i.e., we only assume that the document used for generating each query is relevant and any other document is non-relevant), and we observe statistically significant performance drop in nearly all cases. In the second ablation study, we exclude the seed document generation and use the domain instruction itself as the query to retrieve documents from W and construct the collection C. This leads to an even larger performance drop. Our last ablation focuses on converting the iterative collection construction part to a single retrieval run (i.e., retrieving 10,000 documents in response to the seed document). We observe that in this case, some collections hurt more than others. For example, performance drop on Quora is more significant than FiQA and TREC COVID. Generally speaking, the iterative process leads to a better performance.

Evaluating the Quality of the Synthetic Corpus Construction Approach. To provide a deeper look into the quality of the corpus that we construct in our model, we take the union of W and all the target domain collections listed above. We then ran our synthetic corpus construction experiment to see the accuracy of the model in retrieving the documents that actually belong to the target corpus. We report the average performance in Figure 5.2. In the left plot, we vary the number of generated seeds by ChatGPT and we observe that a single seed document is sufficient and including more documents degrades the accuracy of constructed collection. In the middle plot, we vary the number of retrieved documents per query (i.e., k in Algorithm 1) and observe that the model shows a relatively stable performance compared to various values of k, however, smallest value led to the poorest performance. In the last experiment, we increased the synthetic corpus size from 1,000 to 5,000 and observe that the accuracy of reconstructing document from the actual target domain decreases. However, this performance decrease is not substantial, and the accuracy is still higher than 48% when selecting 5,000 documents. This is another signal to show that the proposed approach for corpus construction performs effectively.

Table 5.6: Attribute-value extraction results for each attribute in our taxonomy. We use ROUGE-L and Exact Match (EM) in addition to manual annotation to evaluate the model. Average results across 15 datasets are reported.

		Instruction Only			Instruction + 1 Example			Instruction + 2 Examples			Instruction + 3 Examples		
	Retrieval Attribute	**ROUGE-L**	**EM**	**Manual**	**ROUGE-L**	**EM**	**Manual**	**ROUGE-L**	**EM**	**Manual**	**ROUGE-L**	**EM**	**Manual**
Query Attributes	Query topic	0.800	0.800	0.800	0.711	0.666	0.733	0.733	0.733	0.733	0.733	0.733	0.733
	Query linguistic features	0.600	0.600	0.600	0.800	0.800	0.800	0.866	0.866	0.866	0.866	0.866	0.866
	Query language	0.666	0.666	0.667	1.000	1.000	1.000	0.866	0.866	0.866	1.000	1.000	1.000
	Query structure	0.099	0.066	0.133	0.866	0.866	0.800	0.933	0.933	0.933	1.000	1.000	1.000
	Query modality	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Query format	0.662	0.533	0.733	0.822	0.733	0.800	0.811	0.733	0.933	0.866	0.866	1.000
Doc Attributes	Document topic	0.666	0.666	0.733	0.733	0.733	0.733	0.733	0.733	0.800	0.800	0.800	0.800
	Document linguistic features	0.800	0.800	0.800	0.800	0.800	0.800	0.866	0.866	0.866	0.933	0.933	0.933
	Document language	0.266	0.266	0.266	0.800	0.800	0.800	0.800	0.800	0.800	0.866	0.866	0.866
	Document structure	0.066	0.066	0.066	0.733	0.733	0.733	0.933	0.933	0.933	1.000	1.000	1.000
	Document modality	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Document format	0.377	0.200	0.533	0.677	0.600	0.866	0.800	0.800	0.867	0.711	0.666	0.800
	Document source	0.836	0.800	0.866	0.826	0.533	0.866	0.893	0.666	0.933	0.933	0.733	0.933
	Relevance notion	0.524	0.133	0.466	0.689	0.533	0.800	0.701	0.533	0.666	0.807	0.733	0.866
	Average	0.454	0.400	0.619	0.818	0.771	0.843	0.852	0.819	0.871	0.894	0.871	0.914

Analyzing the Domain Attribute-Value Extraction Component. As described in Section 5.2.3, we provided three IR experts with all 15 public collections in the BEIR benchmark, and asked them to come up with a description for each retrieval task associated with each collection in a collaborative session. We later presented them with our taxonomy and asked them to annotate the descriptions accordingly. The input of the domain attribute-value extraction model is the task description, in addition to arbitrary choice of examples, and the output is expected to be the value of taxonomy attributes.

Considering we cast the problem of domain attribute-value extraction to a sequence-to-sequence format, following the literature, we used ROUGE-L [95] and Exact Match as our evaluation metrics. ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation) is a commonly used evaluation metric in NLP for summarization tasks, measuring overlap between n-grams in reference summaries and the generated summary. The "L" refers to the longest common subsequence. ROUGE-L scores range from 0 to 1, with 1 being a perfect match. Exact Match (EM) measures the percentage of predictions that exactly match the ground truth, with 1 being a perfect match and 0 no match. Since the task is generative, automatic metrics may not be sufficient, so three annotators manually labeled the outputs of each model, scoring 1 if desirable and 0 if not. Final labels were decided through majority voting.

Table 5.6 presents the results of ChatGPT for domain attribute-value extraction. We made sure that the model is not benefiting from any session data by initiating a new session for each experiment. Each cell displays the average of scores for a particular attribute across 15 collections. The last row reflects the overall performance of each setting based on the average of all attributes. As expected, the highest performance is mostly achieved with Instruction and 3 examples is given. The reason is that the model receives more examples, thus has a better chance of encountering similar cases. As Table 5.6 illustrates, the results of the manual metric highly correlate with the automatic metrics, except for the query and document modality attributes in the instruction only setting. We observe that in this setting, modality attributes resulted in 0.00 with the automatic metrics, but they resulted in 1 in manual annotation. After looking into results, we figured the disparity is because ground truth labels the modality feature as uni-modal, multi-modal, etc. but the sequence-to-sequence model labels it differently, e.g., text. This issue resolves after seeing one example in prompt. We also observe that query and document structure attributes result in a close-to-zero performance in the instruction-only setting. This may be due to the fact that in our instruction, we only provided the model with examples of values for these attributes. However, these attributes have been implicitly mentioned in the domain descriptions, and some in-domain knowledge is necessary to interpret the structure or modality of the task. Again, the performance would significantly improve after seeing only one example. Note that all datasets within BEIR are unstructured, so the model may repeat the only label it has given as an example for structure and modality attributes.

Further, we observe that relevance notion is one of the hardest attributes to predict. This makes sense because usually, understanding what constitutes relevance requires a deep understanding of the task, which these models currently lack. A deep dive into the results showed us that in many cases, the model generalizes the

query attributes to the document attributes, especially in cases that are not explicitly described. For example, if the query topic attribute is predicted as "medical," the model may generalize it to the document topic as well. However, we know that IR features are not necessarily symmetric. A medical query could request information from a heterogeneous corpus such as the Web, and the symmetric assumption makes data synthesis unrealistic.

5.4 Summary

This chapter introduced a new category of domain adaptation methods for neural information retrieval and proposed a pipeline that leverages target domain descriptions to construct a synthetic target collection, generate queries, and produce pseudo-relevant labels. The results of experiments conducted on five diverse target collections demonstrated that our proposed approach outperforms existing dense retrieval baselines in such a domain adaptation scenario. A limitation of this work is that we only collected one description for each retrieval domain, while the performance of the trained IR model can depend on the provided description. Studying and improving the robustness of adaptive ranking models with respect to various description formulations are important avenues to explore in the future.

This work holds the potential for practical applications where the target collection and its relevance labels are unavailable, while preserving privacy and complying with legal restrictions. Future work involves incorporating additional domain-specific information, such as data source and language, and evaluating its conceptualizing ability with more implicit descriptions.

CHAPTER 6

CONCLUSIONS & FUTURE DIRECTIONS

This chapter provides a brief overview of key findings of this dissertation and continues with potential future directions.

6.1 Conclusions and Key Findings

This dissertation leverages retrieval augmentation as a foundational framework, developing efficient and effective retrieval augmented models for information retrieval applications, including query representation, conversational search, and domain adaptation. The research explores the unique challenges posed in IR and solves them with retrieval augmentation, where an auxiliary IR system serves another retrieval system rather than end-users directly. Chapter 3 provides an overview of conversational search, emphasizing the need for accurate representations in user-system conversations and proposing a retrieval-augmented neural network architecture for representing information seeking conversations between a user and a system. The developed solution, Guided Transformer, extends the Transformer architecture [175]–the current state-of-the-art deep learning architecture for representing sequential data, such as natural language text. Experiments on the Qulac dataset [2] suggest that employing Guided Transformer in conversational search models can lead to up to 29% performance improvements in terms of mean reciprocal rank (MRR). Follow-up work from other researchers also demonstrate the effectiveness of Guided Transformer for retrieval augmentation.

Chapter 4 studies the task of learning multiple latent representations for search queries. It challenges the widespread practice of using a single representation for each query, especially for ambiguous and faceted queries. This chapter introduces a novel framework based on retrieval augmentation for learning multiple representations, each representing one of the query intents. This is achieved by end-to-end clustering of retrieval results and learning permutation-invariant representations for the clusters. Extrinsic evaluation via query facet generation tasks demonstrates that the proposed approach can lead to facets with significantly higher quality.

Chapter 5 delves into domain adaptation in IR and introduces a novel domain adaptation task, where the retrieval model adapts to a target domain without accessing its collection directly but relying on a textual description explaining the target domain. We define a taxonomy of domain attributes in retrieval tasks to understand different properties of a source domain that can be adapted to a target domain. We also proposed a pipeline that leverages the target domain description to construct a synthetic target collection, generate queries, and produce pseudo-labels. We then use his synthetic target collection for training a retrieval model for the target domain. Extensive set of experiments on diverse domains, such as biomedical, financial, and scientific domains, demonstrate significant improvements compared to state-of-the-art retrieval baselines for this task. This work holds the potential for practical applications where the target collection and its relevance labels are unavailable, while preserving privacy and complying with legal restrictions.

6.2 Potential Future Directions

***Synergistic End-to-End Learning* for Retrieval-Augmented Representation Learning** Current retrieval-augmented models often use an off-the-shelf retrieval model, often with pre-trained frozen parameters, and combine its output with the input of the downstream network. A promising avenue for future research in the

realm of retrieval-augmented representation learning involves the exploration of end-to-end optimizations, where the parameters of both retrieval and downstream models are updated simultaneously. This integrated approach not only has the potential to enhance the constructive interaction between the two models but also allows for a more cohesive representation learning framework, addressing the challenges of interdependence and mutual reinforcement. Investigating the dynamics of end-to-end training in the context of retrieval-augmented representation learning could pave the way for more robust and efficient models, advancing the capabilities of information retrieval systems in various downstream applications.

***Efficient* Representation Learning with Retrieval Augmentation** In the rapidly evolving landscape of artificial intelligence, a prominent challenge emerges as contemporary models continue to swell in size. The trajectory of recent advancements has witnessed a trend towards larger and more complex models. While these large networks often yield state-of-the-art performance on a multitude of tasks, their scale presents a formidable challenge. The resource-intensive nature of these models presents difficulties in terms of computational requirements, memory constraints, and environmental impact. These challenges necessitate a re-evaluation of the conventional wisdom that larger models inherently lead to superior performance. One alternative to address the issue could be exploring the optimization and enhancement of models with retrieval augmentation for the sake of making them more efficient without a significant performance drop. For example, the RETRO language model from Google DeepMind [12] used Guided Transformer—a retrieval-augmented mechanism introduced in this dissertation—to achieve GPT3 performance but with 25 times fewer model parameters. It is worth exploring other potential retrieval augmentation solutions to intensify the model knowledge without increasing the number of its parameters.

Tailoring the Exposure of Retrieval Augmentation One promising direction for future research involves investigating how much and in what capacity we expose the downstream model to external information provided by the retrieval module. Currently, researchers often add this extra information as the model's input, changing how the input is represented from the beginning. However, it is worth exploring whether these changes could happen at various stages as the information goes through the different layers of the model. We suggest looking into adjusting the depth of exposure based on task-specific factors, the quality of retrieved information, and the specific goals of augmentation in each downstream task. This approach offers a nuanced way to fine-tune how the main model interacts with retrieved information, potentially improving the overall effectiveness of retrieval-augmented representation learning.

***Communication Protocol* Between the Retrieval and Downstream Models** In the future, we will investigate the communication protocol between the retrieval and downstream models. should the query submitted to the retrieval model be in unstructured text form, or is a structured, semi-structured, or latent representation more suitable? Understanding how the query is generated is crucial – is the query produced by an external model or derived from the downstream model? Further exploration is needed to study mechanisms that filter out unnecessary information for the retrieval process. This step aims to prevent the model from being confounded by imprecise or ambiguous queries.

Equally vital is the consideration of the output from the retrieval model. Should all the top k retrieved documents be augmented, or are there alternative methods to select specific pieces of information? This exploration aims to provide the model with a clearer hint for parameter updates, contributing not only to the effectiveness of the learning process but also optimizing memory and computational resources.

The communication dynamics between the retrieval and downstream models require further investigation. Determining how these two components interact and whether the communication protocols generalize across various tasks will unveil insights critical for refining and advancing retrieval-augmented representation learning. This exploration into communication protocols holds the potential to enhance the adaptability and efficiency of the model in diverse applications.

BIBLIOGRAPHY

[1] Abdul-jaleel, Nasreen, Allan, James, Croft, W. Bruce, Diaz, Fernando, Larkey, Leah, Li, Xiaoyan, Metzler, Donald, Smucker, Mark D., Strohman, Trevor, Turtle, Howard, and Wade, Courtney. Umass at trec 2004: Novelty and hard. In *In Proceedings of TREC '04* (2004).

[2] Aliannejadi, Mohammad, Zamani, Hamed, Crestani, Fabio, and Croft, W. Bruce. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2019), SIGIR'19, ACM, p. 475–484.

[3] Asai, Akari, Schick, Timo, Lewis, Patrick, Chen, Xilun, Izacard, Gautier, Riedel, Sebastian, Hajishirzi, Hannaneh, and Yih, Wen-tau. Task-aware retrieval with instructions, 2022. arXiv preprint.

[4] Attar, R., and Fraenkel, A. S. Local feedback in full-text retrieval systems. *J. ACM 24*, 3 (1977), 397–417.

[5] Aust, H., Oerder, M., Seide, F., and Steinbiss, Volker. Experience with the philips automatic train timetable information system. In *Proceedings of 2nd IEEE Workshop on Interactive Voice Technology for Telecommunications Applications* (10 1994), pp. 67 – 72.

[6] Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization, 2016.

[7] Balaneshinkordan, Saeid, Kotov, Alexander, and Nikolaev, Fedor. Attentive neural architecture for ad-hoc structured document retrieval. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2018), CIKM '18, Association for Computing Machinery, p. 1173–1182.

[8] Barrow, Harry G., Tenenbaum, Jay M., Bolles, Robert C., and Wolf, Helen C. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *IJCAI* (1977), pp. 659–663.

[9] Belkin, Nicholas J. Salton award lecture: People, interacting with information. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015* (2015), Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto, Eds., ACM, pp. 1–2.

[10] Bondarenko, Alexander, Gienapp, Lukas, Fröbe, Maik, Beloucif, Meriem, Ajjour, Yamen, Panchenko, Alexander, Biemann, Chris, Stein, Benno, Wachsmuth, Henning, Potthast, Martin, and Hagen, Matthias. *Overview of Touché 2021: Argument Retrieval*. Springer, Online, 03 2021, pp. 574–582.

[11] Boni, Marco, and Manandhar, Suresh. Implementing clarification dialogue in open-domain question answering. *Natural Language Engineering* (2005), 343–361.

[12] Borgeaud, Sebastian, Mensch, Arthur, Hoffmann, Jordan, Cai, Trevor, Rutherford, Eliza, Millican, Katie, van den Driessche, George, Lespiau, Jean-Baptiste, Damoc, Bogdan, Clark, Aidan, de Las Casas, Diego, Guy, Aurelia, Menick, Jacob, Ring, Roman, Hennigan, Tom, Huang, Saffron, Maggiore, Loren, Jones, Chris, Cassirer, Albin, Brock, Andy, Paganini, Michela, Irving, Geoffrey, Vinyals, Oriol, Osindero, Simon, Simonyan, Karen, Rae, Jack W., Elsen, Erich, and Sifre, Laurent. Improving language models by retrieving from trillions of tokens, 2021.

[13] Boteva, Vera, Gholipour, Demian, Sokolov, Artem, and Riezler, Stefan. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval* (Cham, 2016), Springer International Publishing, pp. 716–722.

[14] Braslavski, Pavel, Savenkov, Denis, Agichtein, Eugene, and Dubatovka, Alina. What do you mean exactly? analyzing clarification questions in cqa. In *CHIIR '17* (2017).

[15] Burges, Christopher J. C. From ranknet to lambdarank to lambdamart: An overview. Tech. rep., Microsoft Research, 2010.

[16] Cai, Fei, and de Rijke, Maarten. A survey of query auto completion in information retrieval. *Found. Trends Inf. Retr. 10*, 4 (2016), 273–363.

[17] Campos, Daniel Fernando, Nguyen, Tri, Rosenberg, Mir, Song, Xia, Gao, Jianfeng, Tiwary, Saurabh, Majumder, Rangan, Deng, Li, and Mitra, Bhaskar. Ms marco: A human generated machine reading comprehension dataset. *30th Conference on Neural Information Processing Systems, NIPS* (2016).

[18] Choi, Eunsol, He, He, Iyyer, Mohit, Yatskar, Mark, Yih, Wen-tau, Choi, Yejin, Liang, Percy, and Zettlemoyer, Luke. QuAC: Question answering in context. In *EMNLP '18* (Brussels, Belgium, 2018), pp. 2174–2184.

[19] Christakopoulou, Konstantina, Radlinski, Filip, and Hofmann, Katja. Towards conversational recommender systems. In *KDD '16* (2016), p. 815–824.

[20] Clarke, Charles L.A., Craswell, Nick, and Soboroff, Ian. Overview of the trec 2009 web track. In *TREC '09* (2009).

[21] Coden, Anni, Gruhl, Daniel, Lewis, Neal, and Mendes, Pablo N. Did you mean a or b? supporting clarification dialog for entity disambiguation. In *SumPre '15* (2015).

[22] Cohan, Arman, Feldman, Sergey, Beltagy, Iz, Downey, Doug, and Weld, Daniel. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 2270–2282.

[23] Cool, Colleen, Stein, Adelheit, and Thiel, Ulrich. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications 9* (02 1970), 379–395.

[24] Cormack, Gordon V., Smucker, Mark D., and Clarke, Charles L. Efficient and effective spam filtering and re-ranking for large web datasets. *Inf. Retr. 14*, 5 (2011), 441–465.

[25] Craswell, Nick, Mitra, Bhaskar, Yilmaz, Emine, and Campos, Daniel. Overview of the trec 2020 deep learning track. In *TREC* (2020).

[26] Croft, Bruce, Metzler, Donald, and Strohman, Trevor. *Search Engines: Information Retrieval in Practice*, 1st ed. Addison-Wesley Publishing Company, USA, 2009.

[27] Croft, W. B., and Harper, D. J. Using Probabilistic Models of Document Retrieval Without Relevance Information. *J. of Documentation 35*, 4 (1979), 285–295.

[28] Croft, W. B., and Thompson, R. H. I3r: A new approach to the design of document retrieval systems. *JASIS* (1987), 389–404.

[29] Croft, W. Bruce. Salton award lecture - information retrieval and computer science: An evolving relationship. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (New York, NY, USA, 2003), SIGIR '03, Association for Computing Machinery, p. 2–3.

[30] Croft, W. Bruce. The importance of interaction for information retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019), SIGIR'19, Association for Computing Machinery, p. 1–2.

[31] Dakka, Wisam, and Ipeirotis, Panagiotis G. Automatic extraction of useful facet hierarchies from text databases. *2008 IEEE 24th International Conference on Data Engineering* (2008), 466–475.

[32] Dalton, Jeffrey, Xiong, Chenyan, and Callan, Jamie. Trec cast 2019: The conversational assistance track overview. In *TREC '19* (2019).

[33] Das, Rajarshi, Godbole, Ameya, Kavarthapu, Dilip, Gong, Zhiyu, Singhal, Abhishek, Yu, Mo, Guo, Xiaoxiao, Gao, Tian, Zamani, Hamed, Zaheer, Manzil, and McCallum, Andrew. Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 113–118.

[34] De Boni, Marco, and Manandhar, Suresh. An analysis of clarification dialogue for question answering. In *NAACL '03* (2003), p. 48–55.

[35] Dehghani, Mostafa, Abnar, Samira, and Kamps, Jaap. The Healing Power of Poison: Helpful Non-relevant Documents in Feedback. In *CIKM '16* (2016).

[36] Dehghani, Mostafa, Azarbonyad, Hosein, Kamps, Jaap, and de Rijke, Maarten. Learning to transform, combine, and reason in open-domain question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019), pp. 681–689.

[37] Dehghani, Mostafa, Rothe, Sascha, Alfonseca, Enrique, and Fleury, Pascal. Learning to attend, copy, and generate for session-based query suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (New York, NY, USA, 2017), CIKM '17, ACM, p. 1747–1756.

[38] Dehghani, Mostafa, Zamani, Hamed, Severyn, Aliaksei, Kamps, Jaap, and Croft, W. Bruce. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2017), SIGIR '17, Association for Computing Machinery, p. 65–74.

[39] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Minneapolis, Minnesota, 2019), NAACL '19, ACL, pp. 4171–4186.

[40] Diaz, Fernando, Mitra, Bhaskar, and Craswell, Nick. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Berlin, Germany, Aug. 2016), ACL '16, ACL, pp. 367–377.

[41] Dou, Zhicheng, Hu, Sha, Luo, Yulong, Song, Ruihua, and Wen, Ji-Rong. Finding dimensions for queries. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (2011), CIKM '11, p. 1311–1320.

[42] Dou, Zhicheng, Jiang, Zhengbao, Hu, Sha, Wen, Ji-Rong, and Song, Ruihua. Automatically mining facets for queries from their search results. *IEEE Trans. on Knowl. and Data Eng. 28*, 2 (2016), 385–397.

[43] Ganguly, Debasis, Roy, Dwaipayan, Mitra, Mandar, and Jones, Gareth J.F. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2015), SIGIR '15, ACM, pp. 795–798.

[44] Gao, Luyu, Ma, Xueguang, Lin, Jimmy, and Callan, Jamie. Precise zero-shot dense retrieval without relevance labels, 2022.

[45] Guo, Jiafeng, Fan, Yixing, Ai, Qingyao, and Croft, W. Bruce. A deep relevance matching model for ad-hoc retrieval. In *CIKM '16* (2016), p. 55–64.

[46] Guo, Jiafeng, Fan, Yixing, Pang, Liang, Yang, Liu, Ai, Qingyao, Zamani, Hamed, Wu, Chen, Croft, W. Bruce, and Cheng, Xueqi. A deep look into neural ranking models for information retrieval. *Information Processing & Management 57*, 6 (2020), 102067.

[47] Guu, Kelvin, Lee, Kenton, Tung, Zora, Pasupat, Panupong, and Chang, Ming-Wei. REALM: retrieval-augmented language model pre-training. *CoRR* (2020).

[48] Hand, DJ. Classifier technology and the illusion of progress. *STATISTICAL SCIENCE 21* (2006), 1–14.

[49] Hartigan, JA, and Wong, MA. Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics* (1979), 100–108.

[50] Hashemi, Helia, Zamani, Hamed, and Croft, W. Bruce. Performance prediction for non-factoid question answering. In *ICTIR '19* (2019), p. 55–58.

[51] Hashemi, Helia, Zamani, Hamed, and Croft, W. Bruce. Guided transformer: Leveraging multiple external sources for representation learning in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2020), ACM, p. 1131–1140.

[52] Hashemi, Helia, Zamani, Hamed, and Croft, W. Bruce. Learning multiple intent representations for search queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), Association for Computing Machinery, p. 669–679.

[53] Hashemi, Helia, Zamani, Hamed, and Croft, W. Bruce. Stochastic optimization of text set generation for learning multiple query intent representations. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2022), CIKM '22, Association for Computing Machinery, p. 4003–4008.

[54] Hashemi, Helia, Zhuang, Yong, Kothur, Sachith Sri Ram, Prasad, Srivas, Meij, Edgar, and Croft, W. Bruce. Dense retrieval adaptation using target domain description. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval* (New York, NY, USA, 2023), ICTIR '23, Association for Computing Machinery, p. 95–104.

[55] He, Ben, and Ounis, Iadh. Finding Good Feedback Documents. In *CIKM '09* (2009), pp. 2011–2014.

[56] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR '16* (2016), pp. 770–778.

[57] He, Yunlong, Tang, Jiliang, Ouyang, Hua, Kang, Changsung, Yin, Dawei, and Chang, Yi. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (New York, NY, USA, 2016), CIKM '16, ACM, p. 1443–1452.

[58] Hemphill, Charles T., Godfrey, John J., and Doddington, George R. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language* (1990).

[59] Hoogeveen, Doris, Verspoor, Karin M., and Baldwin, Timothy. Cqadupstack: A benchmark data set for community question-answering research. In *Proceedings of the 20th Australasian Document Computing Symposium* (2015), ADCS '15, Association for Computing Machinery.

[60] Huang, Po-Sen, He, Xiaodong, Gao, Jianfeng, Deng, Li, Acero, Alex, and Heck, Larry. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (New York, NY, USA, 2013), CIKM '13, ACM, p. 2333–2338.

[61] Ilse, Maximilian, Tomczak, Jakub M., and Welling, Max. Attention-based deep multiple instance learning. *CoRR* (2018).

[62] Izacard, Gautier, Caron, Mathilde, Hosseini, Lucas, Riedel, Sebastian, Bojanowski, Piotr, Joulin, Armand, and Grave, Edouard. Towards unsupervised dense information retrieval with contrastive learning. *CoRR* (2021).

[63] Izacard, Gautier, and Grave, Edouard. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282* (2020).

[64] Jardine, N., and van Rijsbergen, C.J. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval 7*, 5 (1971), 217–240.

[65] Järvelin, Kalervo, and Kekäläinen, Jaana. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst. 20*, 4 (2002), 422–446.

[66] Jarvelin, Kalervo P. Salton award keynote: Information interaction in context. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), SIGIR '18, Association for Computing Machinery, p. 1–2.

[67] Karpukhin, Vladimir, Oguz, Barlas, Min, Sewon, Lewis, Patrick, Wu, Ledell, Edunov, Sergey, Chen, Danqi, and Yih, Wen-tau. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 6769–6781.

[68] Kassner, Nora, and Schütze, Hinrich. BERT-kNN: Adding a kNN search component to pretrained language models for better QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online, Nov. 2020), Association for Computational Linguistics, pp. 3424–3430.

[69] Khandelwal, Urvashi, Levy, Omer, Jurafsky, Dan, Zettlemoyer, Luke, and Lewis, Mike. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations* (2020).

[70] Khattab, Omar, Potts, Christopher, and Zaharia, Matei. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (2020).

[71] Khattab, Omar, and Zaharia, Matei. 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* (New York, NY, USA, 2020), ACM, p. 39–48.

[72] Kiesel, Johannes, Bahrami, Arefeh, Stein, Benno, Anand, Avishek, and Hagen, Matthias. Toward voice query clarification. In *SIGIR '18* (2018), p. 1257–1260.

[73] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations* (2015), ICLR '15.

[74] Kohlschütter, Christian, Chirita, Paul-Alexandru, and Nejdl, Wolfgang. Using link analysis to identify aspects in faceted web search, 2006.

[75] Komeili, Mojtaba, Shuster, Kurt, and Weston, Jason. Internet-augmented dialogue generation. *CoRR abs/2107.07566* (2021).

[76] Kong, Weize, and Allan, James. Extracting query facets from search results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2013), SIGIR '13, ACM, p. 93–102.

[77] Kong, Weize, and Allan, James. Extending faceted search to the general web. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (2014), CIKM '14, p. 839–848.

[78] Kong, Weize, and Allan, James. Precision-oriented query facet extraction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016), CIKM '16, p. 1433–1442.

[79] Kosiorek, Adam R., Kim, Hyunjik, and Rezende, Danilo J. Conditional set generation with transformers. *CoRR* (2020).

[80] Kuhn, Harold W. *Naval Research Logistics Quarterly*, 1–2 (1955), 83–97.

[81] Kurland, Oren, and Lee, Lillian. Clusters, language models, and ad hoc information retrieval. *ACM Trans. Inf. Syst. 27*, 3 (May 2009).

[82] Kuzi, Saar, Shtok, Anna, and Kurland, Oren. Query expansion using word embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (New York, NY, USA, 2016), CIKM '16, ACM, pp. 1929–1932.

[83] Lafferty, John, and Zhai, Chengxiang. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01* (2001), p. 111–119.

[84] Latha, K., Veni, K. R., and Rajaram, R. Afgf: An automatic facet generation framework for document retrieval. In *2010 International Conference on Advances in Computer Engineering* (2010), pp. 110–114.

[85] Lau, Tessa, and Horvitz, Eric. Patterns of search: Analyzing and modeling web query refinement. In *Proceedings of the Seventh International Conference on User Modeling* (Berlin, Heidelberg, 1999), UM '99, Springer-Verlag, p. 119–128.

[86] Lavrenko, Victor, and Croft, W. Bruce. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2001), SIGIR '01, ACM, pp. 120–127.

[87] Lee, Juho, Lee, Yoonho, Kim, Jungtaek, Kosiorek, Adam R., Choi, Seungjin, and Teh, Yee Whye. Set transformer. *CoRR* (2018).

[88] Lee, Kenton, Chang, Ming-Wei, and Toutanova, Kristina. Latent retrieval for weakly supervised open domain question answering. *CoRR* (2019).

[89] Lewis, Mike, Liu, Yinhan, Goyal, Naman, Ghazvininejad, Marjan, Mohamed, Abdelrahman, Levy, Omer, Stoyanov, Veselin, and Zettlemoyer, Luke. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (July 2020), pp. 7871–7880.

[90] Lewis, Patrick S. H., Perez, Ethan, Piktus, Aleksandra, Petroni, Fabio, Karpukhin, Vladimir, Goyal, Naman, Küttler, Heinrich, Lewis, Mike, Yih, Wen-tau, Rocktäschel, Tim, Riedel, Sebastian, and Kiela, Douwe. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (2020), Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, Eds.

[91] Li, Chengkai, Yan, Ning, Roy, Senjuti B., Lisham, Lekhendro, and Das, Gautam. Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia. In *Proceedings of the 19th International Conference on World Wide Web* (2010), WWW '10, p. 651–660.

[92] Li, Jiwei, and Jurafsky, Dan. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), ACL, pp. 1722–1732.

[93] Li, Ying, Zheng, Zijian, and Dai, Honghua (Kathy). Kdd cup-2005 report: Facing a great challenge. *SIGKDD Explor. Newsl. 7*, 2 (Dec. 2005), 91–99.

[94] Li, Yizhi, Liu, Zhenghao, Xiong, Chenyan, and Liu, Zhiyuan. *More Robust Dense Retrieval with Contrastive Dual Learning*. Association for Computing Machinery, New York, NY, USA, 2021, p. 287–296.

[95] Lin, Chin-Yew. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* (Barcelona, Spain, July 2004), Association for Computational Linguistics, pp. 74–81.

[96] Lin, Jimmy, Nogueira, Rodrigo, and Yates, Andrew. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467* (2020).

[97] Lin, Zhouhan, Feng, Minwei, dos Santos, Cícero Nogueira, Yu, Mo, Xiang, Bing, Zhou, Bowen, and Bengio, Yoshua. A structured self-attentive sentence embedding. *CoRR* (2017).

[98] Liu, Xiaoyong, and Croft, W. Bruce. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2004), SIGIR '04, ACM, p. 186–193.

[99] Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, and Stoyanov, Veselin. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint* (2019).

[100] Locatello, Francesco, Weissenborn, Dirk, Unterthiner, Thomas, Mahendran, Aravindh, Heigold, Georg, Uszkoreit, Jakob, Dosovitskiy, Alexey, and Kipf, Thomas. Object-centric learning with slot attention. *CoRR* (2020).

[101] Lopez-Paz, David, Nishihara, Robert, Chintala, Soumith, Schölkopf, Bernhard, and Bottou, Léon. Discovering causal signals in images. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (2017), CVPR '17, IEEE Computer Society, pp. 58–66.

[102] Lurcock, Pont, Vlugter, Peter, and Knott, Alistair. A framework for utterance disambiguation in dialogue. In *ALTA '04* (2004), pp. 101–108.

[103] Lv, Yuanhua, and Zhai, ChengXiang. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (New York, NY, USA, 2009), CIKM '09, ACM, pp. 1895–1898.

[104] Lv, Yuanhua, and Zhai, ChengXiang. Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2010), SIGIR '10, pp. 579–586.

[105] Lv, Yuanhua, and Zhai, ChengXiang. Revisiting the divergence minimization feedback model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (2014), CIKM '14, pp. 1863–1866.

[106] Maia, Macedo, Handschuh, Siegfried, Freitas, André, Davis, Brian, McDermott, Ross, Zarrouk, Manel, and Balahur, Alexandra. Www'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018* (Republic and Canton of Geneva, CHE, 2018), WWW '18, International World Wide Web Conferences Steering Committee, p. 1941–1942.

[107] Metzler, Donald, and Bruce Croft, W. Linear feature-based models for information retrieval. *Information Retrieval 10*, 3 (Jun 2007), 257–274.

[108] Metzler, Donald, and Croft, W. Bruce. A markov random field model for term dependencies. In *SIGIR '05* (2005), p. 472–479.

[109] Meyerson, A. Online facility location. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science* (2001), pp. 426–431.

[110] Miao, Jun, Huang, Jimmy Xiangji, and Ye, Zheng. Proximity-based Rocchio's Model for Pseudo Relevance. In *SIGIR '12* (2012), pp. 535–544.

[111] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems* (USA, 2013), NIPS'13, Curran Associates Inc., pp. 3111–3119.

[112] Mitra, Bhaskar, and Craswell, Nick. An introduction to neural information retrieval. *Found. Trends Inf. Retr. 13*, 1 (dec 2018), 1–126.

[113] Mitra, Bhaskar, Diaz, Fernando, and Craswell, Nick. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2017), WWW '17, International World Wide Web Conferences Steering Committee, p. 1291–1299.

[114] Montazeralghaem, Ali, Zamani, Hamed, and Shakery, Azadeh. Axiomatic analysis for improving the log-logistic feedback model. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2016), SIGIR '16, ACM, pp. 765–768.

[115] Mostafazadeh, Nasrin, Misra, Ishan, Devlin, Jacob, Mitchell, Margaret, He, Xiaodong, and Vanderwende, Lucy. Generating natural questions about an image. In *ACL '16* (2016), pp. 1802–1813.

[116] Muandet, Krikamol, Balduzzi, David, and Schölkopf, Bernhard. Domain generalization via invariant feature representation, 2013.

[117] Neelakantan, Arvind, Shankar, Jeevan, Passos, Alexandre, and McCallum, Andrew. Efficient non-parametric estimation of multiple embeddings per word in vector space. *CoRR* (2015).

[118] Nogueira, Rodrigo. From doc2query to doctttttquery, 2019. arXiv preprint.

[119] Nogueira, Rodrigo, and Cho, Kyunghyun. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Copenhagen, Denmark, Sept. 2017), ACL, pp. 574–583.

[120] Nogueira, Rodrigo, and Cho, Kyunghyun. Passage re-ranking with BERT. *CoRR* (2019).

[121] Nogueira, Rodrigo Frassetto, and Cho, Kyunghyun. Passage re-ranking with BERT. *CoRR* (2019).

[122] Oddy, Robert N. Information retrieval through man-machine dialogue. *Journal of Documentation* (1977), 1–14.

[123] Oliva, Junier, Poczos, Barnabas, and Schneider, Jeff. Distribution to distribution regression. In *Proceedings of the 30th International Conference on Machine Learning* (2013).

[124] OpenAI. Aligning language models to follow instructions, 2023.

[125] Ouyang, Long, Wu, Jeff, Jiang, Xu, Almeida, Diogo, Wainwright, Carroll L., Mishkin, Pamela, Zhang, Chong, Agarwal, Sandhini, Slama, Katarina, Ray, Alex, Schulman, John, Hilton, Jacob, Kelton, Fraser, Miller, Luke, Simens, Maddie, Askell, Amanda, Welinder, Peter, Christiano, Paul, Leike, Jan, and Lowe, Ryan. Training language models to follow instructions with human feedback, 2022.

[126] Padigela, Harshith, Zamani, Hamed, and Croft, W. Bruce. Investigating the successes and failures of bert for passage re-ranking, 2019.

[127] Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (USA, 2002), ACL '02, ACL, p. 311–318.

[128] Parikh, Ankur P., Täckström, Oscar, Das, Dipanjan, and Uszkoreit, Jakob. A decomposable attention model for natural language inference. *CoRR* (2016).

[129] Park, Dae Hoon, and Chiba, Rikio. A neural language model for query auto-completion. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2017), SIGIR '17, ACM, p. 1189–1192.

[130] Paulus, Romain, Xiong, Caiming, and Socher, Richard. A deep reinforced model for abstractive summarization. *CoRR* (2017).

[131] Peltonen, Jaakko, Strahl, Jonathan, and Floréen, Patrik. Negative relevance feedback for exploratory search with visual interactive intent modeling. In *IUI '17* (2017), p. 149–159.

[132] Pennington, Jeffrey, Socher, Richard, and Manning, Christopher. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (Doha, Qatar, Oct. 2014), EMNLP '14, ACL, pp. 1532–1543.

[133] Pérez-Iglesias, Joaquín, and Araujo, Lourdes. Standard deviation as a query hardness estimator. In *SPIRE '10* (2010).

[134] Plank, Barbara. Domain adaptation for parsing, 2011. arXiv preprint.

[135] Plank, Barbara. What to do about non-standard (or non-canonical) language in NLP. *CoRR* (2016).

[136] Ponte, J.M., and Croft, W.B. A language modeling approach to information retrieval. In *SIGIR '98* (1998), pp. 275–281.

[137] Prakash, Prafull, Killingback, Julian, and Zamani, Hamed. *Learning Robust Dense Retrieval Models from Incomplete Relevance Labels*. Association for Computing Machinery, New York, NY, USA, 2021, p. 1728–1732.

[138] Qu, Chen, Yang, Liu, Chen, Cen, Qiu, Minghui, Croft, W. Bruce, and Iyyer, Mohit. *Open-Retrieval Conversational Question Answering*. ACM, New York, NY, USA, 2020, p. 539–548.

[139] Qu, Chen, Yang, Liu, Croft, W. Bruce, Zhang, Yongfeng, Trippas, Johanne R., and Qiu, Minghui. User intent prediction in information-seeking conversations. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval* (New York, NY, USA, 2019), CHIIR '19, Association for Computing Machinery, p. 25–33.

[140] Qu, Chen, Yang, Liu, Qiu, Minghui, Zhang, Yongfeng, Chen, Cen, Croft, W. Bruce, and Iyyer, Mohit. Attentive history selection for conversational question answering. In *CIKM '19* (2019), p. 1391–1400.

[141] Qu, Chen, Zamani, Hamed, Yang, Liu, Croft, W. Bruce, and Learned-Miller, Erik. *Passage Retrieval for Outside-Knowledge Visual Question Answering*. Association for Computing Machinery, New York, NY, USA, 2021, p. 1753–1757.

[142] Radford, Alec, Narasimhan, Karthik, Salimans, Tim, and Sutskever, Ilya. Improving language understanding by generative pre-training, 2018. arXiv preprint.

[143] Radlinski, Filip, and Craswell, Nick. A theoretical framework for conversational search. In *CHIIR '17* (2017).

[144] Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, and Liu, Peter J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

[145] Rao, Sudha, and Daumé III, Hal. Answer-based Adversarial Training for Generating Clarification Questions. In *NAACL '19* (2019).

[146] Reddy, Siva, Chen, Danqi, and Manning, Christopher D. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics 7* (2019), 249–266.

[147] Reimers, Nils, and Gurevych, Iryna. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 3982–3992.

[148] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. Okapi at trec–3. In *Proceedings of the Third Text REtrieval Conference (TREC–3)* (1995), pp. 109—126.

[149] Robertson, Stephen, and Zaragoza, Hugo. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[150] Rocchio, J. J. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*. 1971, pp. 313–323.

[151] Ruthven, Ian, and Lalmas, Mounia. A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev. 18*, 2 (2003), 95–145.

[152] Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. *Commun. ACM 18*, 11 (Nov. 1975), 613–620.

[153] Salton, Gerard, Fox, Edward A., and Wu, Harry. Extended boolean information retrieval. Tech. rep., Cornell University, USA, 1982.

[154] Sanderson, Mark. Ambiguous queries: Test collections need more sense. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2008), SIGIR '08, ACM, p. 499–506.

[155] Sanh, Victor, Webson, Albert, Raffel, Colin, Bach, Stephen H., Sutawika, Lintang, Alyafeai, Zaid, Chaffin, Antoine, Stiegler, Arnaud, Scao, Teven Le, Raja, Arun, Dey, Manan, Bari, M Saiful, Xu, Canwen, Thakker, Urmish, Sharma, Shanya Sharma, Szczechla, Eliza, Kim, Taewoon, Chhablani, Gunjan, Nayak, Nihal, Datta, Debajyoti, Chang, Jonathan, Jiang, Mike Tian-Jian, Wang, Han, Manica, Matteo, Shen, Sheng, Yong, Zheng Xin, Pandey, Harshit, Bawden, Rachel, Wang, Thomas, Neeraj, Trishala, Rozen, Jos, Sharma, Abheesht, Santilli, Andrea, Fevry, Thibault, Fries, Jason Alan, Teehan, Ryan, Bers, Tali, Biderman, Stella, Gao, Leo, Wolf, Thomas, and Rush, Alexander M. Multitask prompted training enables zero-shot task generalization, 2021.

[156] Schedl, Markus, Zamani, Hamed, Chen, Ching-Wei, Deldjoo, Yashar, and Elahi, Mehdi. Current challenges and visions in music recommender systems research. *Int. J. Multim. Inf. Retr. 7*, 2 (2018), 95–116.

[157] Sepliarskaia, Anna, Kiseleva, Julia, Radlinski, Filip, and de Rijke, Maarten. Preference elicitation as an optimization problem. In *RecSys '18* (2018), p. 172–180.

[158] Shen, Xuehua, and Zhai, ChengXiang. Active Feedback in Ad Hoc Information Retrieval. In *SIGIR '05* (2005), pp. 59–66.

[159] Shi, Baoguang, Bai, Song, Zhou, Zhichao, and Bai, Xiang. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters* (2015).

[160] Snell, Jake, Swersky, Kevin, and Zemel, Richard S. Prototypical networks for few-shot learning. *CoRR* (2017).

[161] Stoica, Emilia, Hearst, Marti, and Richardson, Megan. Automating creation of hierarchical faceted metadata structures. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics* (2007), pp. 244–251.

[162] Stoyanchev, Svetlana, Liu, Alex, and Hirschberg, Julia. Towards natural clarification questions in dialogue systems. In *AISB '14* (2014).

[163] Su, Hang, Maji, Subhransu, Kalogerakis, Evangelos, and Learned-Miller, Erik G. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision* (2015), ICCV '15, IEEE Computer Society, pp. 945–953.

[164] Sukhbaatar, Sainbayar, Szlam, Arthur, Weston, Jason, and Fergus, Rob. End-to-end memory networks, 2015.

[165] Sun, Si, Qian, Yingzhuo, Liu, Zhenghao, Xiong, Chenyan, Zhang, Kaitao, Bao, Jie, Liu, Zhiyuan, and Bennett, Paul. Meta adaptive neural ranking with contrastive synthetic supervision. *CoRR* (2020).

[166] Sun, Yueming, and Zhang, Yi. Conversational recommender system. In *SIGIR '18* (2018), p. 235–244.

[167] Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2014), NIPS'14, MIT Press, p. 3104–3112.

[168] Szabó, Zoltán, Sriperumbudur, Bharath K., Póczos, Barnabás, and Gretton, Arthur. Learning theory for distribution regression. *J. Mach. Learn. Res. 17*, 1 (jan 2016), 5272–5311.

[169] Tay, Yi, Tran, Vinh Q, Dehghani, Mostafa, Ni, Jianmo, Bahri, Dara, Mehta, Harsh, Qin, Zhen, Hui, Kai, Zhao, Zhe, Gupta, Jai, et al. Transformer memory as a differentiable search index. *arXiv preprint arXiv:2202.06991* (2022).

[170] Teevan, Jaime, Dumais, Susan, and Gutt, Zachary. Challenges for supporting faceted search in large, heterogeneous corpora like the web. In *HCIR 2008* (2008).

[171] Thakur, Nandan, Reimers, Nils, Rücklé, Andreas, Srivastava, Abhishek, and Gurevych, Iryna. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)* (2021).

[172] Thoppilan, Romal, Freitas, Daniel De, Hall, Jamie, Shazeer, Noam, Kulshreshtha, Apoorv, Cheng, Heng-Tze, Jin, Alicia, Bos, Taylor, Baker, Leslie, Du, Yu, Li, YaGuang, Lee, Hongrae, Zheng, Huaixiu Steven, Ghafouri, Amin, Menegali, Marcelo, Huang, Yanping, Krikun, Maxim, Lepikhin, Dmitry, Qin, James, Chen, Dehao, Xu, Yuanzhong, Chen, Zhifeng, Roberts, Adam, Bosma, Maarten, Zhou, Yanqi, Chang, Chung-Ching, Krivokon, Igor, Rusch, Will, Pickett, Marc, Meier-Hellstern, Kathleen, Morris, Meredith Ringel, Doshi, Tulsee, Santos, Renelito Delos, Duke, Toju, Soraker, Johnny, Zevenbergen, Ben, Prabhakaran, Vinodkumar, Diaz, Mark, Hutchinson, Ben, Olson, Kristen, Molina, Alejandra, Hoffman-John, Erin, Lee, Josh, Aroyo, Lora, Rajakumar, Ravi, Butryna, Alena, Lamm, Matthew, Kuzmina, Viktoriya, Fenton, Joe, Cohen, Aaron, Bernstein, Rachel, Kurzweil, Ray, Aguera-Arcas, Blaise, Cui, Claire, Croak, Marian, Chi, Ed, and Le, Quoc. Lamda: Language models for dialog applications, 2022.

[173] Trienes, Jan, and Balog, Krisztian. Identifying unclear questions in community question answering websites. In *ECIR '19* (2019).

[174] Trippas, Johanne R., Spina, Damiano, Cavedon, Lawrence, Joho, Hideo, and Sanderson, Mark. Informing the design of spoken conversational search: Perspective paper. In *CHIIR '18* (2018), p. 32–41.

[175] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Ł ukasz, and Polosukhin, Illia. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[176] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, and Polosukhin, Illia. Attention is all you need. In *NeurIPS '17* (2017).

[177] Vijayakumar, Ashwin K., Cogswell, Michael, Selvaraju, Ramprasaath R., Sun, Qing, Lee, Stefan, Crandall, David J., and Batra, Dhruv. Diverse beam search for improved description of complex scenes. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence* (2018), AAAI Press, pp. 7371–7379.

[178] Vinyals, Oriol, Bengio, Samy, and Kudlur, Manjunath. Order matters: Sequence to sequence for sets, 2016.

[179] Vinyals, Oriol, Fortunato, Meire, and Jaitly, Navdeep. Pointer networks. In *Advances in Neural Information Processing Systems* (2015), C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc.

[180] Voorhees, Ellen M. The cluster hypothesis revisited. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 1985), SIGIR '85, ACM, p. 188–196.

[181] Voorhees, Ellen M., Alam, Tasmeer, Bedrick, Steven, Demner-Fushman, Dina, Hersh, William R., Lo, Kyle, Roberts, Kirk, Soboroff, Ian, and Wang, Lucy Lu. TREC-COVID: constructing a pandemic information retrieval test collection. *CoRR* (2020).

[182] Wachsmuth, Henning, Syed, Shahbaz, and Stein, Benno. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2018), pp. 241–251.

[183] Wadden, David, Lin, Shanchuan, Lo, Kyle, Wang, Lucy Lu, van Zuylen, Madeleine, Cohan, Arman, and Hajishirzi, Hannaneh. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), Association for Computational Linguistics, pp. 7534–7550.

[184] Walker, Marilyn A., Passonneau, Rebecca, and Boland, Julie E. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *ACL '01* (Toulouse, France, 2001), pp. 515–522.

[185] Wang, Kexin, Thakur, Nandan, Reimers, Nils, and Gurevych, Iryna. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Seattle, United States, July 2022), Association for Computational Linguistics, pp. 2345–2360.

[186] Wang, Lucy Lu, Lo, Kyle, Chandrasekhar, Yoganand, Reas, Russell, Yang, Jiangjiang, Eide, Darrin, Funk, Kathryn, Kinney, Rodney, Liu, Ziyang, Merrill, William, Mooney, Paul, Murdick, Dewey A., Rishi, Devvret, Sheehan, Jerry, Shen, Zhihong, Stilson, Brandon, Wade, Alex D., Wang, Kuansan, Wilhelm, Chris, Xie, Boya, Raymond, Douglas, Weld, Daniel S., Etzioni, Oren, and Kohlmeier, Sebastian. CORD-19: the covid-19 open research dataset. *CoRR* (2020).

[187] Wang, Sida, Guo, Weiwei, Gao, Huiji, and Long, Bo. *Efficient Neural Query Auto Completion*. ACM, New York, NY, USA, 2020, p. 2797–2804.

[188] Wang, Xuanhui, Fang, Hui, and Zhai, ChengXiang. A study of methods for negative relevance feedback. In *SIGIR '08* (2008), p. 219–226.

[189] Wang, Yizhong et al. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Abu Dhabi, United Arab Emirates, Dec. 2022), Association for Computational Linguistics, pp. 5085–5109.

[190] Wei, Jason, Bosma, Maarten, Zhao, Vincent Y., Guu, Kelvin, Yu, Adams Wei, Lester, Brian, Du, Nan, Dai, Andrew M., and Le, Quoc V. Finetuned language models are zero-shot learners. *CoRR* (2021).

[191] Welleck, Sean, Kulikov, Ilia, Roller, Stephen, Dinan, Emily, Cho, Kyunghyun, and Weston, Jason. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[192] Weston, Jason, Chopra, Sumit, and Bordes, Antoine. Memory networks, 2014.

[193] Wolf, Thomas, Debut, Lysandre, Sanh, Victor, Chaumond, Julien, Delangue, Clement, Moi, Anthony, Cistac, Pierric, Rault, Tim, Louf, Rémi, Funtowicz, Morgan, and Brew, Jamie. Huggingface's transformers: State-of-the-art natural language processing. *CoRR abs/1910.03771* (2019).

[194] Wu, Wei, Wu, Yu, Xu, Can, and Li, Zhoujun. Knowledge enhanced hybrid neural network for text matching. In *AAAI '18* (2018).

[195] Xiong, Lee, Xiong, Chenyan, Li, Ye, Tang, Kwok-Fung, Liu, Jialin, Bennett, Paul N., Ahmed, Junaid, and Overwijk, Arnold. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations* (2021), ICLR'21.

[196] Xu, Jinxi, and Croft, W. Bruce. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 1996), SIGIR '96, ACM, pp. 4–11.

[197] Xue, Xiaobing, and Croft, W. Bruce. Modeling reformulation using query distributions. *ACM Trans. Inf. Syst.* (May 2013).

[198] Yang, Bishan, and Mitchell, Tom. Leveraging knowledge bases in lstms for improving machine reading, 2019.

[199] Yang, Bo, Wang, Sen, Markham, Andrew, and Trigoni, Niki. Attentional aggregation of deep feature sets for multi-view 3d reconstruction. *CoRR* (2018).

[200] Yang, Liu, Ai, Qingyao, Guo, Jiafeng, and Croft, W. Bruce. Anmm: Ranking short answer texts with attention-based neural matching model. In *CIKM '16* (2016), p. 287–296.

[201] Yang, Liu, Qiu, Minghui, Qu, Chen, Chen, Cen, Guo, Jiafeng, Zhang, Yongfeng, Croft, W. Bruce, and Chen, Haiqing. *IART: Intent-Aware Response Ranking with Transformers in Information-Seeking Conversation Systems*. ACM, New York, NY, USA, 2020, p. 2592–2598.

[202] Yang, Liu, Qiu, Minghui, Qu, Chen, Guo, Jiafeng, Zhang, Yongfeng, Croft, W. Bruce, Huang, Jun, and Chen, Haiqing. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *SIGIR '18* (2018).

[203] Yang, Liu, Zamani, Hamed, Zhang, Yongfeng, Guo, Jiafeng, and Croft, W. Bruce. Neural matching models for question retrieval and next question prediction in conversation. In *NeuIR '17* (2017).

[204] Ye, Zheng, Huang, Jimmy Xiangji, and Lin, Hongfei. Finding a Good Query-Related Topic for Boosting Pseudo-Relevance Feedback. *J. Assoc. Inf. Sci. Technol. 62*, 4 (2011), 748–760.

[205] Zaheer, Manzil, Kottur, Satwik, Ravanbakhsh, Siamak, Póczos, Barnabás, Salakhutdinov, Ruslan, and Smola, Alexander J. Deep sets. *CoRR* (2017).

[206] Zamani, Hamed, Bendersky, Michael, Wang, Xuanhui, and Zhang, Mingyang. Situational context for ranking in personal search. In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2017), WWW '17, International World Wide Web Conferences Steering Committee, p. 1531–1540.

[207] Zamani, Hamed, and Craswell, Nick. Macaw: An extensible conversational information seeking platform. In *SIGIR '20* (2020).

[208] Zamani, Hamed, and Croft, W. Bruce. Embedding-based query language models. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (New York, NY, USA, 2016), ICTIR '16, ACM, pp. 147–156.

[209] Zamani, Hamed, and Croft, W. Bruce. Estimating embedding vectors for queries. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (New York, NY, USA, 2016), ICTIR '16, ACM, p. 123–132.

[210] Zamani, Hamed, and Croft, W. Bruce. Relevance-based word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2017), SIGIR '17, ACM, pp. 505–514.

[211] Zamani, Hamed, and Croft, W. Bruce. On the theory of weak supervision for information retrieval. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval* (New York, NY, USA, 2018), ICTIR '18, ACM, pp. 147–154.

[212] Zamani, Hamed, and Croft, W. Bruce. On the theory of weak supervision for information retrieval. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval* (New York, NY, USA, 2018), ICTIR '18, Association for Computing Machinery, p. 147–154.

[213] Zamani, Hamed, Dehghani, Mostafa, Croft, W. Bruce, Learned-Miller, Erik, and Kamps, Jaap. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2018), CIKM '18, ACM, p. 497–506.

[214] Zamani, Hamed, Diaz, Fernando, Dehghani, Mostafa, Metzler, Donald, and Bendersky, Michael. Retrieval-enhanced machine learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2022), SIGIR '22, Association for Computing Machinery, p. 2875–2886.

[215] Zamani, Hamed, Dumais, Susan, Craswell, Nick, Bennett, Paul, and Lueck, Gord. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020* (New York, NY, USA, 2020), WWW '20, ACM, p. 418–428.

[216] Zamani, Hamed, Lueck, Gord, Chen, Everest, Quispe, Rodolfo, Luu, Flint, and Craswell, Nick. Mimics: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2020), CIKM '20, ACM, p. 3189–3196.

[217] Zamani, Hamed, Mitra, Bhaskar, Chen, Everest, Lueck, Gord, Diaz, Fernando, Bennett, Paul N., Craswell, Nick, and Dumais, Susan T. Analyzing and learning from user interactions for search clarification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2020), ACM, p. 1181–1190.

[218] Zeng, Hansi, Zamani, Hamed, and Vinay, Vishwa. Curriculum learning for dense retrieval distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2022), SIGIR '22, Association for Computing Machinery, p. 1979–1983.

[219] Zhai, ChengXiang. I3A: an intelligent interactive information agent model for information retrieval. In *ICTIR '22: The 2022 ACM SIGIR International Conference on the Theory of Information Retrieval, Madrid, Spain, July 11 - 12, 2022* (2022), Fabio Crestani, Gabriella Pasi, and Éric Gaussier, Eds., p. 2.

[220] Zhai, Chengxiang, and Lafferty, John. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management* (New York, NY, USA, 2001), CIKM '01, ACM, pp. 403–410.

[221] Zhang, Hongfei, Song, Xia, Xiong, Chenyan, Rosset, Corby, Bennett, Paul N., Craswell, Nick, and Tiwary, Saurabh. Generic intent representation in web search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2019), SIGIR'19, ACM, p. 65–74.

[222] Zhang, Tianyi, Kishore, Varsha, Wu, Felix, Weinberger, Kilian Q., and Artzi, Yoav. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (2020), OpenReview.net.

[223] Zhang, Yan, Hare, Jonathon S., and Prügel-Bennett, Adam. Deep set prediction networks. *CoRR* (2019).

[224] Zhang, Yongfeng, Chen, Xu, Ai, Qingyao, Yang, Liu, and Croft, W. Bruce. Towards conversational search and recommendation: System ask, user respond. In *CIKM '18* (2018), p. 177–186.

[225] Zhu, Fengbin, Lei, Wenqiang, Wang, Chao, Zheng, Jianming, Poria, Soujanya, and Chua, Tat-Seng. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774* (2021).

[226] Zhu, Peide, and Hauff, Claudia. Unsupervised domain adaptation for question generation with DomainData selection and self-training. In *Findings of the Association for Computational Linguistics: NAACL 2022* (2022), Association for Computational Linguistics, pp. 2388–2401.