



University of
Massachusetts
Amherst

Context-Aware Query and Document Representation in Information Retrieval Systems

Item Type	Dissertation (Open Access)
Authors	Naseri, Shahrzad
DOI	10.7275/55161
Rights	Attribution 4.0 International
Download date	2026-04-21 22:45:33
Item License	http://creativecommons.org/licenses/by/4.0/
Link to Item	https://hdl.handle.net/20.500.14394/55161

**CONTEXT-AWARE QUERY AND DOCUMENT
REPRESENTATION IN INFORMATION RETRIEVAL
SYSTEMS**

A Dissertation Presented

by

SHAHRZAD NASERI

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

DOCTOR OF PHILOSOPHY

September 2024

Manning College of Information and Computer Sciences

© Copyright by Shahrzad Naseri 2024

All Rights Reserved

**CONTEXT-AWARE QUERY AND DOCUMENT
REPRESENTATION IN INFORMATION RETRIEVAL
SYSTEMS**

A Dissertation Presented

by

SHAHRZAD NASERI

Approved as to style and content by:

James Allan, Chair

W. Bruce Croft, Member

Mohit Iyyer, Member

Jeffrey Dalton, Outside Member

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

ACKNOWLEDGMENTS

I am deeply grateful to my advisor, James Allan, for his invaluable guidance, unwavering support, and patience throughout my PhD journey. His encouragement and understanding were particularly comforting during the difficult times I encountered. His vast knowledge and extensive experience have been a constant source of inspiration, both in my academic research and personal life. I could not have asked for a better advisor and mentor.

I extend my heartfelt thanks to my thesis committee: Bruce Croft, Mohit Iyyer, and Jeff Dalton, for their insightful comments and encouragement. Their challenging questions pushed me to expand my research from different angles. I am especially grateful to Jeff for his detailed feedback and invaluable input during our close collaboration, as well as for hosting me at the University of Glasgow during my scholar visit in the summer of 2019. It has been a privilege to work with such a distinguished group of experts.

I have been fortunate to have outstanding mentors during my summer internships at Spotify, Microsoft, and Netflix. Working closely with Sravana Reddy, Joana Correia, Jussi Karlgren, and Rosie Jones at Spotify; Lingling Zheng, Tushar Kanakagiri, Brent Jensen, and Nick Swanson at Microsoft; and Noble Kennamer, Vito Ostuni, and Sudarshan Lamkhede at Netflix has significantly influenced my approach to research. These experiences not only deepened my understanding of real-world challenges but also greatly boosted my confidence as a researcher.

I am grateful to my fellow labmates and colleagues for the stimulating discussions, the late nights spent working together before deadlines, and the many memorable

moments we've shared over the past few years. I would especially like to thank Zhiqi Huang, John Foley, Hamed Rezanejad, Sheikh Sarwar, Negin Rahimi, Ali Montazer, Youngwoo Kim, Myung-ha Jang, and Lakshmi Vikraman. Additionally, I extend my thanks to the CIIR staff—Jean Joyce, Kate Morruzzi, Dan Parker, Glenn Stowell, and Gregory Brooks—for their invaluable administrative support.

I am profoundly grateful to my parents, Hossein and Mahshid, whose unwavering love and support have been the foundation of my journey. My mother, despite facing a challenging illness, was a constant source of inspiration and strength throughout my PhD. Her enduring belief in my potential, along with my father's steadfast encouragement, gave me the confidence to pursue and achieve my goals. Their countless sacrifices and belief in me are reflected in every accomplishment I have made.

Finally, I extend my deepest gratitude to my partner, Milad Nasresfahani, whose constant support and belief in me were essential in bringing this thesis to completion. His encouragement made this achievement possible.

I am deeply thankful to Pegah Taheri and Soha Rostaminia for their ongoing support, kindness, and friendship, which have been a source of strength throughout this journey. Their presence has uplifted me in countless ways.

I also want to thank my dear friends Alireza Bahramali, Sadegh Rabiee, Hamed Rezanejad, Razieh Faghihpirayesh, Anna Saeedi, Saeed Goodarzi, Amirhossein Ghafari, Ali Montazer, and Hamid Mozafari.

I dedicate this dissertation to my family and friends.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by National Science Foundation (NSF) grant #IIS-1617408, in part by NSF grant #1819477, and in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007 under Univ. of Southern California subcontract no. 124338456. Any views and conclusions contained herein are those of the authors

and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

ABSTRACT

CONTEXT-AWARE QUERY AND DOCUMENT REPRESENTATION IN INFORMATION RETRIEVAL SYSTEMS

SEPTEMBER 2024

SHAHRZAD NASERI

B.Sc., AMIRKABIR UNIVERSITY OF TECHNOLOGY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

Input representation has a major impact on the effectiveness of Information Retrieval (IR) systems. Further, developing a context-aware input representation for IR systems is crucial to answering user's complicated information need. The goal of this work is to take advantage of the *contextual features* to represent the query and document to enhance the information retrieval systems performance. We focus on three sources of *contextual features*: 1. Entities, defined as things or concepts that exist in the world; 2. Context within pseudo-relevant feedback document in IR systems; and 3. Context within example documents provided by user as the IR system's input.

We first introduce a dense entity representation based on the relationships between an entity and other entities described within its summary. We explore its use in the entity ranking task by representing both queries and documents using this

model. By integrating this ranking methodology with a term-based ranking method, we achieved statistically significant improvements over the term-based ranking approach. Further, we developed a retrieval model that merges term-based language model retrieval, word-based embedding ranking, and entity-based embedding ranking, resulting in the best performance. Additionally, we introduce an entity-based query expansion framework employing local and global entity knowledge sources; i.e. corpus-based indexed entities and the summary-expanded entity embedding. Our results demonstrate our entity-based expansion framework outperforms the learned combination of word-based expansion techniques.

Then we focus on leveraging the context of pseudo-relevance feedback documents (PRF) for ranking relevant terms to the user’s query. To achieve this, we utilize transformer models, which excel at capturing context through their attention mechanisms, and expand the query with top-ranked terms. We propose both unsupervised and supervised frameworks. Our unsupervised model employs transformer-generated embeddings to calculate the similarity between a term (from a PRF document) and the query, while considering the term’s context within the document. Our results demonstrate that this unsupervised approach outperforms static embedding-based expansion models and performs competitively with state-of-the-art word-based feedback models, relevance model variants, across multiple collections. The supervised framework approaches query expansion as a binary classification task, aiming to identify terms within the PRF documents relevant to the query. We utilize transformer models in a cross-attention architecture to predict relevancy scores for candidate terms. This supervised approach yields performance comparable to term frequency-based feedback models, relevance model variant. Moreover, combining it with the relevance model results in even greater improvement than either model used independently.

Finally, we concentrate on leveraging the context of the example documents provided by the user in the query-by-example retrieval problem to formulate a latent

query that represents the user’s information needs. We construct three query-by-example datasets and develop several transformer-based re-ranking architectures. Our Passage Relevancy Representation by Multiple Examples (PRRIME) overcomes BERT’s context window limitations by segmenting query example and candidate documents into passages. It then trains an end-to-end neural ranking architecture to aggregate passage-level relevance representations, demonstrating improvement over the first-stage ranking framework. Additionally, we explore a cross-encoder reranking architecture using the Longformer transformer model for query-by-example retrieval, aiming to capture cross-text relationship, particularly aligning or linking matching information elements across documents. This shows statistically significant improvement on the test set of the dataset which it is trained on but performs not as well as the baseline on the other two datasets which have limited fine-tuning data, indicating limited knowledge transferability. Finally, we investigate a dual-encoder reranking architecture that learns query and document representations through an auxiliary training paradigm. It uses query prediction as an auxiliary task alongside the ranking objective as the main task. It outperforms both the initial retrieval stage and the single-loss training method - i.e training the dual encoders solely with a ranking objective.

CONTENTS

	Page
ACKNOWLEDGMENTS	iv
ABSTRACT	vii
LIST OF TABLES	xiii
LIST OF FIGURES	xvi
 CHAPTER	
1. INTRODUCTION	1
1.1 Entities as the Contextual Information Source	3
1.2 PRF Documents as Contextual Information Source	5
1.3 Example Documents as the Contextual Information Source	7
2. RELATED WORK	10
2.1 Query Expansion	10
2.1.1 Relevance Feedback Expansion Models	11
2.1.2 Dense and Sparse Embedding-based Expansion	12
2.1.3 Supervised and End-to-End Expansion Models	13
2.1.4 Dense Retrieval-Based Expansion Models	15
2.1.5 Generative Expansion Models	15
2.2 Leveraging Entities in Information Retrieval Systems	17
2.2.1 Entity-centric Ranking	17
2.2.1.1 Entity Retrieval	17
2.2.1.2 Knowledge Base-Focused Ad-hoc Document Retrieval	19
2.2.1.3 Neural and Embedding Based Approaches	20
2.2.1.4 Complex Entity Centric Queries	22

2.2.1.5	Entity Set Expansion	24
2.3	Text Ranking with Transformers	25
2.4	Query by Example Information Retrieval Systems	28
3.	ENTITIES AS THE CONTEXTUAL INFORMATION SOURCE	30
3.1	Exploring Summary-Expanded Entity Embeddings for Entity Retrieval	32
3.1.1	Summary-Expanded Entity Embeddings	32
3.1.2	General Scheme of Retrieval	33
3.1.3	Experimental Setup	34
3.1.4	Results	35
3.2	Local and Global Query Expansion for Hierarchical Complex Entity-centric Queries	38
3.2.1	Topic Expansion Model	39
3.2.2	Experimental Setup	41
3.2.3	Results	44
3.3	Discussion	46
3.4	Summary	47
4.	PSEUDO RELEVANT FEEDBACK DOCUMENTS AS CONTEXTUAL INFORMATION SOURCE	49
4.1	Unsupervised Query Expansion with Transformers	51
4.1.1	Word and WordPiece representations	51
4.1.2	Contextualized Embeddings for Query Expansion (CEQE)	51
4.2	Supervised Query Expansion with Transformers (SQET)	55
4.3	Experimental Setup	57
4.3.1	Datasets	57
4.3.2	Intrinsic expansion judgments	58
4.3.3	Baselines	59
4.3.3.1	Unsupervised: CEQE	59
4.3.3.2	Supervised: SQET	59
4.3.4	System Details	60
4.4	Results	61

4.4.1	Contextualized Query Expansion	61
4.4.1.1	Comparing CEQE and SQET on Robust	65
4.4.2	PRF effect on Neural Reranking	67
4.4.3	Expansion after Reranking	68
4.4.4	Intrinsic Expansion Evaluation	69
4.5	Qualitative Behavior Analysis	71
4.5.1	Computational Cost Analysis	73
4.6	Discussion	73
4.7	Summary	74
5.	EXAMPLE DOCUMENTS AS THE CONTEXTUAL INFORMATION SOURCE	76
5.1	Query by Example Retrieval Datasets	77
5.2	Neural Re-ranking in Query by Example Retrieval	80
5.2.1	First Stage Retrieval	80
5.3	Passage-based R elevancy R epresentation w I th Multiple E xamples (PRRIME)	81
5.4	Cross-Encoder Reranking in Query Example Retrieval	87
5.5	Multi-task Query Generation and Re-ranking in Query by Example Retrieval	89
5.6	Summary	93
6.	CONCLUSION AND FUTURE WORK	97
6.1	Future Work	100
	BIBLIOGRAPHY	102

LIST OF TABLES

Table	Page
3.1 Query and retrieved entity representations for <code>WORDVEC</code> and <code>ENTITYVEC</code> models.	34
3.2 Effect of <code>WORDVEC</code> and <code>ENTITYVEC</code> models on top of LM baseline for verbose, short queries and their union. Superscripts [†] , [‡] , and [§] indicate statistical significance over the LM, LM+ <code>WORDVEC</code> , and LM+ <code>ENTITYVEC</code> , respectively.	36
3.3 Effect of <code>WORDVEC</code> and <code>ENTITYVEC</code> models on top of LM baseline for different query types. Superscripts [†] , [‡] , and [§] indicate statistical significance over the LM, LM+ <code>WORDVEC</code> , and LM+ <code>ENTITYVEC</code> , respectively.	37
3.4 Effect of <code>WORDVEC</code> and <code>ENTITYVEC</code> models on top of RM3 baseline for verbose, short queries and their union. Superscripts [†] , [‡] , and [§] indicate statistical significance over the RM3, RM3+ <code>WORDVEC</code> , and RM3+ <code>ENTITYVEC</code> , respectively.	37
3.5 Examples of topic expansion features across word and entity vocabularies. All features are for <i>R</i> , <i>I</i> , and <i>H</i> nodes separately. The example topic is: [Antibiotic use in livestock/Use in different livestock/In swine production]. The entities identified in the topic are: [Antibiotics, Livestock/ Livestock/ Domestic pig, Pig farming]	41
3.6 Text-based baselines and expansion methods. * indicates significance over the RH-SDM run.	43
3.7 Learned feature weights of combination of SDM and RM3 over different outline levels using L2R.	43
3.8 Entity-based expansion with varying latent entity models. * indicates significance over the L2R-SDM-RM3 Baseline.	45

4.1	Ranking effectiveness of CEQE on unsupervised baseline retrieval for Deep Learning 2019 Track for the task of full document ranking. The superscript † and ‡ denote statistical significance over BM25 + RM3 and Static-Embed, respectively.	63
4.2	Ranking effectiveness of CEQE on unsupervised baseline retrieval for the Complex Answer Retrieval (CAR) Track. The superscript † and ‡ denote statistical significance over BM25 + RM3 and Static-Embed-PRF, respectively.	64
4.3	Ranking effectiveness on the Robust collection. The superscript † and ‡ denote statistical significance over BM25 + RM3 and Static-Embed-PRF, respectively. Bold indicates the best value in each section of the table.	67
4.4	Ranking effectiveness of neural ranking on top of query expansion methods for Robust. The superscript † and ‡ indicate significance over BM25 + CEDR and (BM25 + RM3) + CEDR with re-ranking the top 1000, respectively. Bold indicates the best value in each section of the table.	67
4.5	Ranking effectiveness of multi-round neural re-ranking and expansion for Robust. The superscript † and ‡ indicate significance over BM25 + CEDR and (BM25 + CEDR) + RM3 baselines, respectively.	69
4.6	Intrinsic ranking evaluation of positive expansion terms on Robust. The superscript † denotes the statistical significance over the Relevance Model. Bold indicates the best result in each column.	71
4.7	Example query expansion terms for Topic [405 , cosmic events] and [685, oscar winner selection] in Robust collection. This includes the important intrinsic positive labels, Relevance Model, CEQE-MAXPool and SQET-Context _{invRank} expansion terms. Terms with positive intrinsic labels are bolded.	72
4.8	Win/Loss comparison to BM25 on Robust.	73
5.1	Statistics of proposed QBE datasets. Avg #d ⁺ /q denotes the average number of relevant documents per query.	79

5.2	Query extraction methods for the first stage retrieval. For each column, the highest value is marked with bold text. At this stage, we select R@100 as the primary evaluation metric. Subscripts refer to the standard deviation of 5 corpuses.	81
5.3	PRRIME Model performance on QBE datasets. Subscripts refer to the standard deviation of 5 corpuses. For PRRIME-Adhoc, statistically significant improvements are marked by ★ (over Keyphrase), ▲ (over PRRIME-Summ).	87
5.4	Cross-encoder reranking results on QBE datasets. Subscripts refer to the standard deviation of 5 corpuses. Statistically significant improvements of CD-Longformer are marked by ★ over Keyphrase.	89
5.5	Performance of Auxiliary training paradigm for query by example retrieval task on Wiki-QBE dataset. Subscripts refer to the standard deviation of 5 corpuses. For Aux-Loss-Rerank, statistically significant improvements are marked by ★ (over Keyphrase), ▲ (over Single-Loss-Rerank).	93
5.6	96

LIST OF FIGURES

Figure	Page
3.1 Example of a complex topic from the TREC Complex Answer Retrieval track.....	39
5.1 Overview of the PRRIME model.	84
5.2 Neural reranking using cross-encoder architecture in query by example retrieval	88
5.3 Overview of the multi-task learning framework.	90

CHAPTER 1

INTRODUCTION

Input representation is a fundamental task in information retrieval (IR), significantly affecting the effectiveness of retrieving relevant documents in response to a user's query. Traditionally, IR models relied on techniques like bag-of-words, where query and documents are represented as an unordered collection of terms, often with frequency weighting (e.g., TF-IDF). These models operate based on the exact matching between high-dimensional vector representation of the queries and the document. As a result, their performance is limited by semantic discrepancies and vocabulary gaps and they are unable to capture the context. To enhance semantic understanding, subsequent research incorporated additional contextual information, particularly entities (Lee, Fuxman, Zhao, & Lv, 2015). Entities which are defined as distinct concepts or objects that exist in the world facilitate the semantic understanding of user intent and provides context for interpreting user's information need.

Further, the advent of deep learning and neural networks in Natural Language Processing (NLP) has transformed input representation. Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013) pioneered dense vector representations (embeddings) that capture word meanings and relationships. While a significant step forward, these embeddings remain static, ignoring the specific context in which words appear.

Transformer architecture (Vaswani et al., 2017) have addressed this limitation by using attention mechanisms to dynamically model context within text. More recently, large-scale Transformer-based language models (LLMs) have achieved state-of-the-art results in text ranking. Researchers have investigated LLMs as both initial

retrievers (Ma, Wang, Yang, Wei, & Lin, 2023; Pradeep, Sharifymoghaddam, & Lin, 2023; Sun et al., 2023) and as pointwise or listwise rerankers. Reranking can be framed as text generation, with models outputting an ordered list (Ma, Zhang, Pradeep, & Lin, 2023) or deriving rankings by sorting token probabilities (Ma, Zhang, et al., 2023).

In this thesis, spanning work from 2018 to 2024, we investigate *context-aware* representations for queries and documents to improve the semantic understanding of query and documents. We focus on three core problems: entity retrieval, query expansion to improve ad-hoc document retrieval performance and query by example retrieval. We emphasize three primary sources of context:

- **Knowledge Base Entities:** Leveraging entities and their relationships with each other from knowledge base.
- **Pseudo-Relevance Feedback (PRF) Document Context:** Utilizing context within documents initially retrieved for the original query which are considered pseudo-relevant to the user’s query.
- **Example Document Context:** Incorporating context from user-provided example documents in the query by example retrieval task.

More specifically, we begin by investigating the impact of entities as a source of additional context for understanding user information needs with the goal of enhancing the IR system performance. For instance, in the query “Population of New York City” recognizing “New York City” as a single entity (a city) rather than isolated words leads to improved IR results compared to a word-level representation alone.

Further, unlike the query “Population of New York City” where the user’s information need is direct and simple and the user is looking for a precise fact, a user’s information need can be complex, consisting of different interconnected parts and aspects. For example answering the query “What are the causes of the Civil War”

can consist of different aspects such as economical, political and social. Taking advantage of different entities and words that are related to these different aspects will provide more context for enriching the query and formulating a context-aware query representation.

Next, we explore how to leverage context within PRF (Pseudo-Relevance Feedback) documents in the query expansion task. We investigate the potential and challenges of using contextual language models (i.e., Transformer models) to represent this context. Our approach includes both unsupervised and supervised methods. The unsupervised method builds upon Relevance Model expansion techniques (Lavrenko & Croft, 2001), while the supervised method frames query expansion as a classification task.

Lastly, we focus on the query by example retrieval problem where users don't provide an explicit query. Instead, they provide example documents, and the IR system aims to find other relevant documents that reflect the underlying information need. We explore and investigate the strengths and challenges of using Transformer models for query and document representation, as these models excel at understanding the context within example documents, helping to represent the latent query they represent.

In Chapter 2, we survey key research areas that provide the foundation for this thesis and review related works developed after our own. These include but are not limited to query expansion techniques, entity ranking, Transformer models for text ranking, and query by example retrieval. This background sets the stage for our contributions, which we outline in Section 1.1, Section 1.2, and Section 1.3.

1.1 Entities as the Contextual Information Source

In Chapter 3, we study the task of *Entity Retrieval* defined as retrieving a ranked list of entities for a given query, and utilize the *contextual information* latent in en-

tities’ characteristics such as their relationship with other entities to represent the queries and documents. In entity retrieval, the user is looking for specific entities as the information need in contrast to only a list of web pages addressing the query. Lastly, the information representing the entities include other entities where there exist structured information about them such as names, aliases, categories etc. We build an entity retrieval system by learning a joint word-entity dense embedding representation that leverages the summary of entity articles from the Wikipedia knowledge base with a focus on mentions of the related entities. The intuition behind using this approach is that the summary of an entity has mentions of important related entities and can be used for enriching query and document representation.

Next, we investigate a context-aware query representation for multifaceted entity-centric complex queries that include hierarchical information. The TREC CAR (Complex Answer Retrieval) dataset provides semi-structured entity-bearing queries that are constructed to address a complex information need. These queries are complex in the sense that they cannot be answered with fact-like answers and the answers are typically long and multi-faceted. As discussed earlier, the answer to the query “What are the causes of the Civil War?” is multi-faceted and incorporates different aspects from slavery to Lincoln’s election. We study latent word-based and entity-based representations and extend these approaches to complex queries using fine-grained representation on different elements of the hierarchical query structure. We formulate a context-aware query representation by leveraging the learnt word-entity representation mentioned above and enrich the queries with contextual information derived from similar entities and words. This results in improving the recall of probabilistic retrieval approaches. Since this model incorporates universal information about the entities we refer to it as global expansion method. Also, we investigate the local models derived from pseudo-relevance feedback expansion approaches. Our main contributions are:

Contribution 1.1: *We introduce a simple entity embedding model that focuses on representing an entity based on other entities crucial to its summary with the goal of incorporating the contextual information of the entities relationship with each other in the embedding representation. We demonstrate that utilizing the entity-based representation results in 5.4% improvement in MAP@1000 over the Query Likelihood (QL) retrieval model for all queries comparing to an only word-based embedding representation that results in 3.9% improvement over QL.*

Contribution 1.2: *We develop entity-aware query expansion methods based on probabilistic retrieval approaches and entity embedding vectors for passage retrieval given complex, multifaceted, and hierarchical queries. We show a mixture model of different entity-based expansion model, capturing both global context (embedding representation) and local context (retrieving entities within the corpus) outperforms a learned combination of probabilistic word-based models by 21%.*

These studies were conducted from 2017 to 2019 and pre-date recent advances in NLP, particularly Transformer models and large language models.

1.2 PRF Documents as Contextual Information Source

In Chapter 4, we investigate the use of latent embedding vectors generated by pre-trained contextual language models like BERT (Devlin, Chang, Lee, & Toutanova, 2019) for query expansion and enriching query representation. These models, trained on massive datasets, are recognized for their ability to encode linguistic and factual knowledge within their deep neural network parameters.

Further, with their attention architecture they enable us to utilize the context in which a term occurs to enhance the query representation. We build a new model,

Contextualized Embeddings for Query Expansion (CEQE), based on pseudo-relevance feedback. Previously, most pseudo-relevance feedback models operated at the word level without considering the word’s context. CEQE utilizes query-focused contextualized embedding vectors and the contextualized embedding vectors of terms in the pseudo-relevant documents to find the best terms for unsupervised expansion of the query. Further, we design a supervised query expansion model, Supervised Contextualized Query Expansion with Transformers (SQET), that builds on the Transformer based architecture and treats the expansion problem as a supervised classification task leveraging the context words around the candidate expansion terms.

Contribution 2.1: *We develop a new contextualized query expansion method, CEQE, that shifts from word-count approaches to contextualized approaches. Our results demonstrate that CEQE significantly outperforms static embedding expansion methods in terms of Mean Average Precision (MAP) by 23% on TREC Deep Learning 19 and 7% on TREC Robust04 datasets. CEQE also achieves comparable performance to static embeddings on the TREC CAR dataset. Furthermore, CEQE statistically significantly surpasses a variant word-based relevance feedback model, RM3, which combines relevance feedback with query expansion by interpolating the original query with terms from feedback documents, by 4% in MAP on the TREC Deep Learning 19 dataset, while maintaining competitive results on the TREC Robust04 and TREC CAR datasets.*

Contribution 2.2: *We demonstrate that rounds of neural re-ranking, query expansion using CEQE and a final neural re-ranking outperforms single round of neural re-ranking in terms of MAP@100 by 8%.*

Contribution 2.3: *We develop a supervised query expansion model, SQET, by formulating query expansion as a classification task leveraging Transformer-based models in a cross-attention architecture. The linear combination of the*

SQET model and the RM3 word-based relevance feedback model results in a statistically significant performance gain. Specifically, MAP and Recall both improve by 2% when compared to using the RM3 model alone demonstrating distinct contribution of SQET to retrieval performance.

1.3 Example Documents as the Contextual Information Source

In Chapter 5, we study the task of Query by Example (QBE), where the user provides one or more documents representing their information need, and the IR system aims to retrieve a list of additional relevant documents. The task of QBE is challenging and complex because unlike traditional ad-hoc retrieval we don't have a specific information need in the form of a short or even a verbose query and in fact the users' information need is scattered through the example documents.

To study this problem, we construct three QBE datasets derived from standard keyword-based ad hoc document retrieval tasks, including WikIR-QBE (a large-scale dataset for training and evaluation), Robust04-QBE, and MultiNews-QBE (both evaluation datasets). Next, we develop Transformer-based neural re-rankers and address the challenges of integrating Transformer models into Query by Example (QBE) retrieval problem. Our re-rankers leverage the attention architecture of Transformers to provide context-aware representations of queries and documents, optimizing the ranking. In particular, we developed Passage Relevancy Representation with Multiple Examples (PRRIME) which tackles the limited context window of BERT-based models by splitting documents into passages and then trains an end-to-end neural ranking architecture that aggregates passage-level relevance representations. Our results show that PRRIME outperforms the first-stage term-based retrieval methods, which use the top- k ranked terms from TF-IDF (Sparck Jones, 1972) as the query, across all the datasets.

We investigate a cross-encoder architecture leveraging Longformer (Beltagy, Peters, & Cohan, 2020) (a Transformer model designed for long documents at the time of developing this architecture) to re-rank first-stage retrievals in query by example retrieval. To capture interactions, special tokens mark the beginning and end of both example and candidate documents. This architecture allows for analysis of relationships within query example documents as well as between query and candidate documents. Our approach outperforms the first-stage retrieval on the dataset used for its initial fine-tuning. However, subsequent cross-validation fine-tuning on two other evaluation datasets perform not as well as the first-stage retrieval, likely caused by a limited number of available samples and the model being overfit to the characteristics of initial training collection.

Lastly, we developed a dual-encoder architecture that includes an auxiliary query prediction task. This task is designed to enhance the primary ranking objective, resulting in a model that not only generalizes better but also offers explainability for its latent query representations. The model trained under the auxiliary training paradigm outperforms the dual encoder ranker trained only with ranking objective as well as the first-stage retrieval on the WikIR-QBE collection. In summary, our main contributions are:

Contribution 3.1: *We build multiple publicly available query by example retrieval datasets based on the standard keyword-based adhoc document retrieval datasets for training and evaluation, namely WikIR-QBE, Robust04-QBE and MultiNews-QBE.*

Contribution 3.2: *We develop a cross-encoder neural re-ranker architecture, PRRIME, that employ a passage-based relevancy representation between example documents’ passages and candidate document’s passages. Our results show that this model achieve significant improvements of 37% and 19% in*

terms of $MAP@100$ improvement over the baseline, a query likelihood retrieval model using top-ranked TF-IDF n -grams, on the Wiki-QBE and MultiNews-QBE datasets. PRRIME achieves comparable results to the baseline on the Robust04-QBE dataset, though we hypothesize further improvement is limited because of the dataset’s small size.

Contribution 3.3: We investigate the effectiveness of the cross-encoder architecture for reranking in query by example tasks. We introduce special tokens to the Longformer Transformer model, enabling it to better capture relationships between example documents. Our results demonstrate an 18% improvement in Mean Average Precision (MAP) over the baseline, a query likelihood retrieval model using top-ranked TF-IDF n -grams. However, we discover our model is tuned on the latent characteristic of the train dataset and does not perform well on other datasets with limited number of instances.

Contribution 3.4: We build a dual-encoder architecture with an auxiliary query prediction task to enhance the primary ranking objective. This model significantly outperforms the dual encoder ranker that uses only the ranking objective, as well as first-stage retrieval methods on the WikIR-QBE collection and MultiNews-QBE datasets, achieving an improvement of 17% and 10% in MAP, respectively.

Our solutions and findings in this dissertation establish robust baselines for leveraging entities as contextual sources and using Transformer-based models to represent textual context within query and documents in information retrieval tasks. These solutions and findings provide a foundation for future work exploring the potential of the recent large-scale language models such as GPT-3.5, GPT-4, Gemini, Llama, etc.

CHAPTER 2

RELATED WORK

In this chapter, we describe the background and related work essential to understanding this thesis, along with subsequent research that expands upon our findings. First, we discuss the long-standing problem of query expansion which has been used as a technique for providing more context for the user’s information need (Section 2.1). Then, we provide an overview of utilizing knowledge graphs and entities, as a source of contextual information, to enhance information retrieval systems as well as discuss the entity retrieval problem (Section 2.2). Next, we discuss utilizing Transformer-based architecture models such as BERT, Longformer, etc. for the tasks of text ranking, query refinement, and document representation. Subsequently we address the recent developments in utilizing Large Language Models such as GPT3.5, GPT4, Vicuna for ranking and re-ranking. (Section 2.3). Lastly, we discuss work related to the query by example retrieval problem (Section 2.4).

2.1 Query Expansion

One of the fundamental challenges in retrieval is the vocabulary mismatch problem which arises from synonymy (words with similar meaning) and polysemy (words with multiple meanings). Query expansion reformulates and expands the original query with related terms in order to improve effectiveness, recall in particular. We categorize the related work on query expansion into five main sections: 1)Relevance Feedback Expansion Models, 2)Embedding-based Expansion Models, 3)Supervised and End-to-End Expansion Models, 4)Dense Retrieval-based Expansion Models and

5) Generative Expansion Models. These categories are interrelated rather than distinct. Each section often incorporates principles and methodologies from the others, demonstrating a significant overlap and integration of approaches.

2.1.1 Relevance Feedback Expansion Models

Relevance feedback is a powerful technique that leverages user-provided judgments on the relevance of retrieved documents to update the original query formulation. The Rocchio algorithm, a traditional implementation of relevance feedback, updates the query vector by moving it towards the centroid of relevant documents and away from irrelevant ones (Rocchio, 1971). Pseudo-relevance feedback (PRF) (Lavrenko & Croft, 2001; Lv & Zhai, 2009; Zhai & Lafferty, 2001) approaches perform the task of identifying relevant documents automatically, *assuming* the top retrieved documents are relevant. RM3 (Lavrenko & Croft, 2001) is a variant of PRF models which is a mixture model between the top k expansion terms and the original query. Expansion terms are weighted according to their term frequency in the top-ranked high scoring documents.

PRF documents can provide contextual information to enhance a user’s query. One of the earliest studies on utilizing context for query expansion was conducted by (Xu & Croft, 1996), whose Local Context Analysis (LCA) model expands a query using concepts that frequently co-occur with the query terms. These concepts are identified as noun phrases within top-ranked passages relevant to the original query. In our query expansion models introduced in Chapter 4, we leverage Transformer models whose self-attention mechanism and pre-training on massive text corpora enable us to generate a richer contextual representation of expansion terms stemming from the PRF documents. This leads to improved ranking of potential expansion terms, ultimately enhancing the effectiveness of our query expansion process.

2.1.2 Dense and Sparse Embedding-based Expansion

Another approach for query expansion incorporates static embeddings (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Pennington, Socher, & Manning, 2014) to find the terms relevant to the query. Word embedding techniques learn a low-dimensional vector (compared to the vocabulary size) for each vocabulary term in which the similarity between the word vectors captures the semantic as well as the syntactic similarities between the corresponding words. Word embeddings are unsupervised learning methods since they only need raw text data without other explicit labels. There are different methods to compute the word embeddings. One of the most popular methods is using neural networks to predict words based on the context of a text. Mikolov, Sutskever, et al. (2013) introduced Word2Vec that learns vector representation of words via a neural network with a single layer. Word2vec is proposed in two ways, CBOW and skip-gram. CBOW tries to predict the word based on the context, i.e., neighboring words. skip-gram tries to predict the context. Given the word w , it tries to predict the probability of word w' being in a fixed window of word w . Another model for learning embedding vectors is based on matrix factorization, e.g., GloVe vectors (Pennington et al., 2014). Although many variants of word embeddings exist, skipgram embeddings are quite efficient and not significantly different from other variations if tuned correctly (Levy, Goldberg, & Dagan, 2015; Mikolov, Chen, et al., 2013).

Since embeddings promise to capture the semantic similarity between terms, they are used in different ways to expand queries (Diaz, Mitra, & Craswell, 2016; Kuzi, Shtok, & Kurland, 2016; Roy, Paul, Mitra, & Garain, 2016; Zamani & Croft, 2016, 2017). Zamani and Croft (2016) proposed two expansion methods based on unigram language models with either an assumption that query terms are conditionally independent or an assumption that term similarities are query-independent. Similarly, Kuzi et al. (2016) perform expansion using locally trained word embeddings by either

finding the closest terms to the query’s centroid or by identifying the terms with the highest embedding similarity to each query term. In the latter case, these per-term similarities are then aggregated using either a sum, weighted sum, or max over query terms. These word embeddings, such as Word2Vec, GloVe, and others, learn a static word embedding for each term regardless of the context. Most basic models fail to address polysemy and the contextual characteristics of terms.

Both our unsupervised (CEQE) and supervised (SQET) expansion models introduced in Chapter 4 leverage the contextual representation of the terms to expand the query with terms relevant to the query, to address the static representations limitations. Further, other works that leverage term’s contextual representation for term-weighting are query-independent (Dai & Callan, 2020a, 2020b), while our models considers the original query when selecting expansion terms.

Concurrent with our proposed model CEQE, Formal, Piwowarski, and Clinchant (2021) introduced the Sparse Lexical and Expansion Model for First Stage Ranking (SPLADE), which employs Transformer models, specifically BERT, to learn sparse representations of queries and documents. SPLADE utilizes the Masked Language Modeling task to determine the importance of token j in the vocabulary to token i in the input sequence. The final representation of term j is obtained by using a log-saturated function applied to the aggregate of importance scores across the input sequence tokens. To optimize these representations, SPLADE incorporates in-batch negative sampling in their ranking loss calculation and employs sparse regularization techniques.

2.1.3 Supervised and End-to-End Expansion Models

There is a vein of work using supervised learning to perform query expansion. Cao, Nie, Gao, and Robertson (2008) and Imani, Vakili, Montazer, and Shakery (2019) use feature-based models to predict what terms should be used for expansion. A common

practice is to classify terms as positive, negative, or neutral and use classification methods to maximize the number of predicted positive terms. Further, an end-to-end neural PRF model (NPRF) proposed by Li et al. (2018) uses a combination of models to compare document summaries and compute document relevance scores for feedback and achieves limited improvement while only using bag-of-words neural models. Zheng et al. (2020) identify and rank relevant text chunks within the first-pass retrieved documents. They then utilize the top-k chunks as queries, using a cross-encoder to score candidate documents. The obtained scores were then weighted and summed to determine the final relevance score of the candidate documents.

Researchers have studied leveraging Reinforcement Learning (RL) to optimize the selection of terms for query expansion. Nogueira and Cho (2017) introduce a query reformulation system based on a convolutional neural network (CNN) / Recurrent Neural Networks (RNN) that rewrites a query to maximize the number of relevant documents returned. However, their model only focus on semantic matching between query and a candidate term and cannot capture relevance matching signals such as term importance, document frequency of candidate terms in the feedback set and document length. Montazerlghaem, Zamani, and Allan (2020) introduced a reinforcement learning framework for query expansion that directly optimizes retrieval metrics, including average precision for effective retrieval and α -nDCG for diverse retrieval. Moreover, X. Wang, Macdonald, and Ounis (2020) proposed a reinforcement learning-based seq2seq model for query reformulation. The reward function in their RL framework utilizes query performance prediction to select high-quality paraphrases resulting in encouraging the model to focus on paraphrases likely to enhance retrieval effectiveness.

2.1.4 Dense Retrieval-Based Expansion Models

Dense retrieval which encodes queries and documents into high-dimensional vectors using neural learning models to enable semantic similarity and utilizes approximate nearest neighbors search to quickly find the most relevant documents, has shown impressive results (Khattab & Zaharia, 2020; S.-C. Lin, Yang, & Lin, 2020; Qu et al., 2021; L. Xiong et al., 2020). Researchers have studied incorporating pseudo-relevance feedback for dense retrieval to improve query representation. H. Yu, Xiong, and Callan (2021) introduced ANCE-PRF which concatenates the original query with pseudo-relevance feedback (PRF) passages obtained from an ANCE retrieval model (L. Xiong et al., 2020) and encodes the combined text using a BERT architecture. The new query encoder is then trained for dense retrieval ranking, while the document encoder remains fixed. X. Wang, Macdonald, Tonellotto, and Ounis (2021) proposed ColBERT-PRF, a vector-based pseudo-relevance feedback (PRF) technique that enhances ColBERT’s query-document scoring function. In particular, ColBERT-PRF clusters the embeddings of pseudo-relevant documents retrieved using the ColBERT dense retrieval model. It then selects the most discriminative embeddings by mapping them to token space and prioritizing tokens with high inverse document frequency (IDF). Finally, a linear combination of ColBERT’s query-document score and the weighted similarity between selected embeddings and the document’s token-level representation is used to calculate final query-document relevance score. Later they propose a deep language model-based contrastive weighting model, called CWPRF that learns to select the most useful embeddings for expansion (X. Wang, MacAvaney, Macdonald, & Ounis, 2023a).

2.1.5 Generative Expansion Models

Nogueira, Yang, Lin, and Cho (2019) were among the first to leverage Transformer models for generative expansion approaches. Their work focused on generating

questions from passages and concatenating them to the document, falling under the umbrella of “document expansion” methodologies. This approach differs from ours, which investigates query expansion instead. For the TREC Deep Learning 2019 track, the Brown team addressed query expansion by framing it as a query paraphrasing problem (Zerveas, Zhang, Kim, & Eickhoff, 2020). They trained their paraphrasing model, based on a sequence-to-sequence architecture using OpenNMT (Klein, Kim, Deng, Senellart, & Rush, 2017), on the MS MARCO passage dataset (Bajaj et al., 2016). They focused on the 2.6% of passages that answer multiple queries, providing the model with rich examples of how the same information need can be expressed differently. During inference, the model generates paraphrases of the original query; the top 3 outputs from a k-beam search are concatenated with the original query to expand the query. X. Wang, MacAvaney, Macdonald, and Ounis (2023b) investigated neural query reformulation methods built upon small generative neural models, such as T5 and FLAN-T5 models. They proposed two possible generative query reformulation frameworks: GenQR, where models directly take a query as input, and GenPRF, where models also incorporate contextual information extracted from pseudo-relevant feedback documents.

Recent advancements in Large Language Models (LLMs) such as GPT-3.5, GPT-4, Gemini, and Claude have dramatically improved natural language understanding, text generation, and task completion. These advancements allow researchers to directly apply LLMs in downstream tasks (zero-shot settings) or fine-tune them for specialized purposes. In the query expansion problem, Mackie, Chatterjee, and Dalton (2023b) leveraged GPT-3 to generate diverse query-specific text formats (e.g., keywords, entities, chain-of-thought reasoning, facts, news articles, and essays). They demonstrated that combining these generated text types outperforms sparse retrieval techniques, in particular BM25, across multiple datasets. They later showed that combining this approach with traditional PRF expansion techniques results in fur-

ther improvement as their generative expansion techniques and PRF expansion have contrasting benefits (Mackie, Chatterjee, & Dalton, 2023a).

2.2 Leveraging Entities in Information Retrieval Systems

Researchers have explored leveraging entities (specific things and concepts in real world) and knowledge bases (structured repositories of entity information and relationships) for enhancing information retrieval systems. In this thesis, we investigate the use of entities to augment user queries and corpus documents with richer context. In Section 2.2.1, we provide an overview of “Entity-centric Ranking” approaches, including Entity Retrieval, Knowledge Base-Focused Ranking, Neural and Embedding Approaches, Complex Entity Centric queries and Entity Set Expansion.

2.2.1 Entity-centric Ranking

2.2.1.1 Entity Retrieval

Entity ranking is the task of retrieving entities from a knowledge base and presenting them in ranked order in response to a user’s information need. This focus has driven various benchmarking campaigns including the INEX Entity Ranking track (Demartini, Iofciu, & De Vries, 2009), the INEX Linked Data Track (Q. Wang et al., 2012), the TREC Entity track (Balog, Carmel, & Arjen, 2012; Balog, Serdyukov, & Vries, 2010; Serdyukov & De Vries, 2009), the Semantic Search Challenge (Blanco et al., 2011; Halpin et al., 2010), and the Question Answering over Linked Data (QALD) challenge series (Lopez, Unger, Cimiano, & Motta, 2013). These campaigns share the goal of providing an entity-focused response to users, instead of returning documents which might contain unnecessary information. The campaigns differed in the specific types of queries they addressed, such as list searches (Balog et al., 2012; Demartini et al., 2009), related entity finding (Serdyukov & De Vries, 2009), and natural language questions (Lopez et al., 2013). To facilitate further benchmarking of entity ranking

systems, Balog and Neumayer (2013) and Hasibi et al. (2017) compiled datasets from the mentioned campaigns, introducing the DBPedia Entity-V1 and DBPedia Entity-V2 datasets, respectively. We evaluate the effectiveness of our *Summary-Expanded Entity Embeddings* model for entity retrieval on the DBPedia Entity-V2 dataset in Section 3.1.

Many different approaches have been proposed to address the entity retrieval (entity ranking) task. Zhiltsov, Kotov, and Nikolaev (2015) propose a fielded retrieval approach known as the Fielded Sequential Dependence Model (FSDM). This model uses a learning-to-rank method to determine the importance of various entity fields, such as name, attributes, categories, similar entities, and related entities. Further, Hasibi, Balog, and Bratsberg (2016) proposed leveraging entity annotations of queries obtained by entity linking in the entity retrieval model. Schuhmacher, Dietz, and Paolo Ponzetto (2015) propose a learning-to-rank model which incorporates different features of both text and entities for entity ranking. Foley, O’Connor, and Allan (2016) propose an approach for entity ranking that does not rely on entity linking to be effective even with limited linguistic resources that are typically annotated by experts. Additionally, Garigliotti, Hasibi, and Balog (2019) studied the effectiveness of entity type information in entity retrieval.

Further, Dietz, Gamari, and Dalton (2018) introduced the Complex Answer Retrieval dataset, featuring entity-centric, multi-faceted hierarchical queries that address both entity ranking and passage ranking tasks. In Section 2.2.1.4, we provide a detailed description of this dataset and discuss relevant approaches for it. This relates to our *Local and Global Entity Centric model* (Section 3.2), which also investigates these queries.

2.2.1.2 Knowledge Base-Focused Ad-hoc Document Retrieval

Researchers explored utilizing entities within knowledge bases in different ways to improve ad-hoc document retrieval. Dalton, Dietz, and Allan (2014) proposed EQFE model for web-based queries which expand the query with entity features such as alias, category, type. X. Liu, Chen, Fang, and Wang (2014) propose an entity-centric query expansion for enterprise data. They later propose the Latent Entity Space model (X. Liu & Fang, 2015), which constructs entity profiles (pooled information about entity in the document collection or the entity document in the knowledgebase) and represent both query and documents with their entities. C. Xiong and Callan (2015b) propose a query expansion technique by using Freebase knowledge graph objects to improve ad-hoc information retrieval. Further, they propose a learning-to-rank approach that uses *objects* from external data such as vocabularies and entities as an interlingua between query and documents. The learning-to-rank model then learns a weighting for the features that are derived from query-object and object-document relationships (C. Xiong & Callan, 2015a). Later, they propose a bag-of-entity representation for query and documents which ranks documents either by number of query entities that they contain or by the frequency of query entities in the document (C. Xiong, Callan, & Liu, 2016). They further address the discrepancies between entity linking and entity-based ranking systems by performing the two tasks jointly (C. Xiong, Liu, Callan, & Hovy, 2017). Recently, (Shehata, Arabzadeh, & Clarke, 2022) study expanding both queries and documents with entity mentions to improve the performance of sparse retrievers like BM25. While their approach improved on the BM25 baseline on the MSMarco dataset, it still fell short of dense retriever performance. Despite not matching dense retriever performance, the study objective was to bridge the gap between sparse and dense retrievers as dense retrievers have shortcomings such as latency issues.

2.2.1.3 Neural and Embedding Based Approaches

With the rise of dense embedding representations and deep learning, researchers explored training entity-based embeddings and deep neural entity-based models, leveraging them in information retrieval systems. Yamada et al. (2018) propose the Wikipedia2Vec model, which aims to learn a joint representation of words and entities. This is achieved by linearly combining three kinds of context: word context (represented by a Word2Vec approach on Wikipedia articles), entity context (represented by neighboring entities in Wikipedia’s link graph), and anchor context (represented by the words surrounding the anchor text of an entity in Wikipedia’s article).

The entity-based embeddings have been used for many ranking tasks. For example C. Xiong, Power, and Callan (2017) show the effectiveness of academic knowledge graph embeddings for academic search. Later, Gerritse, Hasibi, and de Vries (2020) explore the effectiveness of Wikipedia2vec for representing queries and documents in entity retrieval tasks. They utilize the similarity between query and document vector representations based on Wikipedia2vec as a re-ranking step, following an initial ranking with a fielded probabilistic retrieval model on the DBPedia-Entity V2 dataset.

Further, C. Xiong, Callan, and Liu (2017) propose the word-entity duet model for ad-hoc document retrieval which the query and documents are represented by both word-based and entity-based hand-crafted features. The four-way interactions between the two representation spaces form a word-entity duet that can systematically incorporate various semantics from the knowledge graph. Z. Liu, Xiong, Sun, and Liu (2018) expand on word-entity duet model and instead of using hand-crafted features uses a translation layer that calculates similarity between a pair of query-document terms (words and entities) along with a neural matching ranker. C. Xiong, Liu, Callan, and Liu (2018) propose the end-to-end neural kernel entity salience model which

estimates entity salience (importance) in documents and show it improves the ad-hoc document retrieval performance.

The advent of Transformer models prompted researchers to investigate whether incorporating entity contextual information from a knowledge base could further enhance their performance in downstream tasks. ERNIE (Gerritse, Hasibi, & de Vries, 2022) pioneered the integration of entities into Transformer models. This was achieved by pre-training a BERT-based Transformer where token embeddings for entities are aggregated with their corresponding TransE-based (Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013) entity embeddings. Further, KnowBERT (Peters, Neumann, et al., 2019) enhances BERT by integrating knowledge graph-based entity embedding with BERT by introducing a Knowledge Attention and Recontextualization (KAR) component during pre-training. This component aggregates the embeddings of entities linked from knowledge graphs with the mention-span representations computed from BERT vectors. Unlike ERNIE and KnowBERT, which incorporate factual entity knowledge during pre-training, E-BERT (Poerner, Waltinger, & Schütze, 2020) enhances BERT by aligning Wikipedia2Vec entity vectors with BERT’s wordpiece vectors using a transformation matrix. The aligned entity vector is then concatenated next to the wordpiece representations of the entity mentions. E-BERT has been shown to improve performance on unsupervised QA tasks like LAMA, as well as downstream tasks like relation classification. Gerritse et al. (2022) combined E-BERT with the mono-BERT (Nogueira & Cho, 2019) cross-encoder re-ranking architecture and employed it for entity retrieval task achieving state-of-the-art results for entity-centric queries. Recently, Chatterjee, Mackie, and Dalton (2024) recently proposed an end-to-end re-ranking architecture for ad-hoc document retrieval, which utilizes a hybrid document embedding that combines query-specific entity-centric and text-based embeddings. Their approach outperforms existing non-entity based and entity-based methods on standard TREC collections.

2.2.1.4 Complex Entity Centric Queries

TREC CAR Dataset TREC Complex Answer Retrieval (CAR)(Dietz et al., 2018) is a dataset curated for the TREC Complex Answer Retrieval track to address retrieval for complex entity-centric topics, which was introduced in 2017 (TREC CAR Y1) and continued for 2018 (TREC CAR Y2) and 2019 (TREC CAR Y3). In this dataset each topic consists of the hierarchical skeleton of a Wikipedia article and its sections. To be more specific, [Radiocarbon dating/Measurement and results/Errors and reliability] is an example topic constructed from the hierarchical skeleton of the “Radiocarbon dating¹” Wikipedia article. The TREC CAR dataset defines two tasks: 1) passage ranking, where the goal is to retrieve paragraphs, and 2) entity ranking, where the goal is to retrieve entities for each query. The most common approach to formulate a query from a topic is to concatenate the different parts of its hierarchical skeleton. The TREC CAR setup includes two types of judgments, *automatic* and *manual*. The automatic (binary) judgments are derived directly from Wikipedia and the manual judgments are created by NIST assessors. TREC CAR has different relevance annotations based on the section path of the topic. *Tree-qrel* relevancy judgments which label a paragraph or entity as relevant if it is contained in the section or any of its child sections and *Hierarchical* relevancy judgments which assess the relevance of paragraphs or entities that are directly included only within that specific section. There are 2283 evaluation topics for BenchMarkY1Test for the Tree Qrels.

Comparison of TREC CAR with Other Query Types. TREC CAR queries, while not strictly formulated as traditional questions, share core methodologies with non-factoid Question Answering (QA) (Breja & Jain, 2021; Cohen & Croft, 2016; Cohen, Yang, & Croft, 2018; Cortes, Woloszyn, Barone, Möller, & Vieira, 2021; Song

¹https://en.wikipedia.org/wiki/Radiocarbon_dating

et al., 2017; Vikraman, MontazerAlghaem, Hashemi, Croft, & Allan, 2021). Both approaches rely on retrieving relevant passages to provide comprehensive answers. However, the key distinction lies in the structure and complexity of the queries. CAR topics are multi-faceted with hierarchical relationships that necessitate complex, multi-part answers, whereas non-factoid QA typically deals with singular questions that, although not seeking simple factual answers, do not generally require the hierarchical or multi-dimensional responses characteristic of CAR queries.

On the other hand, complex queries in retrieval is not a new problem. In fact, some of the earliest uses of retrieval focused on boolean retrieval. Users constructed complex boolean expressions with complex subqueries (Salton, Fox, & Wu, 1983). This was later followed up with more complex query capability (Turtle & Croft, 1991). Follow-up query languages that support rich query expressions include: INQUERY, Lucene, Terrier, and Galago. However, these languages are often only used internally to rewrite simple keyword queries, possibly using some inferred structure from natural language processing. In contrast, CAR query topics contain explicit multifaceted hierarchical structure.

Findings on TREC CAR. Nanni, Mitra, Magnusson, and Dietz (2017) survey approaches for the CAR dataset, including sparse retrieval (BM25, TF-IDF), query expansion (Lavrenko & Croft, 2001), dense vector word-based (GloVe (Pennington et al., 2014)) and entity-based (RDF2Vec (Ristoski & Paulheim, 2016)) embedding ranking, supervised learning-to-rank and neural re-ranking, Duet model (Mitra, Diaz, & Craswell, 2017). They find that RDF2Vec entity embeddings are not as effective as BM25 for their entity-focused paragraph ranking, and although the neural Duet model gives the best performance, the gains over BM25 are only modest. Later, MacAvaney et al. (2018) modified the PACRR (Hui, Yates, Berberich, & de Melo, 2017) neural re-ranker for the CAR dataset. They added contextual vector features—*Heading Position* (indicating a term’s location within a heading) and *Heading Usage Frequency*

(similar to Inverse Document Frequency (IDF), measuring a heading’s importance across documents). These modifications improved effectiveness over the unmodified PACCR model and other approaches like BM25 and the Sequential Dependence Model (Metzler & Croft, 2005). With the advent of Transformer models, Nogueira and Cho (2019) introduced the mono-BERT ranking architecture, modeling passage relevance calculation as a sequence classification task. Applied to the TREC-CAR dataset, this approach significantly improved the MAP metric by 2X compared to non-neural and earlier neural baselines

2.2.1.5 Entity Set Expansion

Another vein of work focus on finding “sibling” entities within a corpus that are from the set characterized by a small set of seed entities. P. Yu, Huang, Rahimi, and Allan (2019) introduce an unsupervised corpus-based set expansion framework that leverages lexical features as well as distributed representations of entities. In a follow-up work (P. Yu, Rahimi, Huang, & Allan, 2020), they present a two-channel neural re-ranking model, that jointly learns exact and semantic matching of entity contexts through entity interaction features. The key difference between entity retrieval tasks and entity set expansion lies in the query format. Entity retrieval tasks accept various queries, including natural language questions (e.g., “Who is the mayor of Berlin?”), keyword queries (e.g., “electronic music genres”), named entity queries (e.g., “Brooklyn Bridge”), and requests for specific lists of entities (e.g., “Professional sports teams in Massachusetts”). In contrast, entity set expansion uses an example set of entities (e.g., “Boston Celtics”, “Patriots”, “Boston Bruins”) as the query, with the goal of extracting related entities (e.g., “Boston Red Sox”, “New England Revolution”).

2.3 Text Ranking with Transformers

Transformer-based language models have revolutionized the field of Natural Language Processing (NLP). Pioneered by BERT Devlin et al. (2019), these models leverage the self-attention mechanism to dynamically weigh the importance of tokens (words or subwords) within their input sequence. Through large-scale pre-training on extensive text corpora, Transformer models obtain a rich understanding of linguistic structure and semantic relationships. This facilitates effective transfer learning across diverse NLP tasks, including sentiment analysis, question answering, document ranking, etc. (Akkalyoncu Yilmaz, Yang, Zhang, & Lin, 2019; Dai & Callan, 2019; Li, Yates, MacAvaney, He, & Sun, 2020; MacAvaney, Yates, Cohan, & Goharian, 2019; Nogueira & Cho, 2019; Nogueira, Jiang, & Lin, 2020; Padigela, Zamani, & Croft, 2019; Qiao, Xiong, Liu, & Liu, 2019; H. Zhang et al., 2019).

Recent Large Language Models (LLMs) or Foundational Models leverage substantially scaled architectures and even larger training datasets than those employed by first-generation Transformer-based models like BERT, T5, etc. These models adopt a decoder-only generative approach, producing text one token at a time. LLMs perform remarkably well in zero-shot settings, demonstrating impressive capabilities without direct fine-tuning on specific downstream tasks (Hou et al., 2024; Kojima, Gu, Reid, Matsuo, & Iwasawa, 2022).

In this section, we first provide a brief overview of related work using first-generation Transformer-based models – i.e. BERT-based models, T5, etc. – for document ranking. We then explore how LLMs are applied to ranking, both in zero-shot settings and with downstream task fine-tuning.

At the beginning of 2019 Nogueira and Cho (2019) achieved state-of-the-art results by being the first to propose fine-tuning BERT as a reranker and use it as a binary classifier to predict whether the document is relevant. Later, Qiao et al. (2019) explore and compare various methods for integrating the BERT embeddings

of queries and documents to calculate relevance scores. Their approaches include computing cosine similarity between the [CLS] tokens from the last layer of BERT for both the query and document, as well as concatenating the query and document and using the [CLS] token of the concatenated representation to determine the relevance score

MacAvaney et al. (2019) propose incorporating Transformer models into existing neural ranking architectures such as PACRR (Hui, Yates, Berberich, & De Melo, 2018), KNRM (C. Xiong, Dai, Callan, Liu, & Power, 2017) and DRMM (Guo, Fan, Ai, & Croft, 2016). They achieve this by treating each layer of a contextualized language model as a channel and integrating the channel-specific query-document similarity matrices within the existing ranking architecture.

The above approaches to document ranking with BERT-based Transformer models either focus on datasets with passage-length entries or truncate long documents to fit the maximum input length. Consequently, researchers focused on adapting model architectures to address limited input-sequence length challenge. The works of Akkalyoncu Yilmaz et al. (2019), Li et al. (2020), and Y. Kim, Rahimi, Bonab, and Allan (2021) were among the first to propose segmenting long documents into sentences or passages for independent scoring against a query, with the final document relevance determined by aggregating these scores. In particular, Dai and Callan (2019) score individual passages and aggregate them into document level score by taking the first, maximum, or sum of passages scores. Li et al. (2020) propose to aggregate passage *representations* rather than passage relevance scores. On the other hand, with the advent of Transformer models with larger context-window size researchers study and compare the performance of such models for ranking long documents (Boytsov et al., 2022; Sekulić, Soleimani, Aliannejadi, & Crestani, 2020).

Further, other researchers (Gao, Dai, Fan, & Callan, 2020; Khattab & Zaharia, 2020; MacAvaney et al., 2020; L. Xiong et al., 2021; Zhan, Mao, Liu, Zhang, & Ma,

2020) utilize Transformer models to produce query and document representations that can be used for (relatively) efficient first-stage retrieval. In this context, Gao et al. (2020) find that combining a representation-based model with a lexical matching component improves effectiveness. Zheng et al. (2020) combine BERT with a NPRF framework and illustrate the importance of an effective first-stage ranking method. Padaki, Dai, and Callan (2020) investigate BERT’s performance when using expanded queries and find that expansion which preserves some linguistic structure is preferable to expanding with keywords.

Recently, with the advent of Large Language Models (LLMs) there have been a shift from adopting smaller language models (such as BERT, Longformer, etc.) towards LLMs for ranking and re-ranking. In particular, Ma, Wang, et al. (2023) fine-tuned a LLaMA-2 model (Touvron et al., 2023), employing a bi-encoder architecture (Karpukhin et al., 2020) with a contrastive ranking objective for first-pass retrieval. They further fine-tuned LLaMA-2 within a cross-attention architecture, concatenating query and documents and utilizing the same contrastive ranking objective for a re-ranking stage. On the other hand, other researchers cast re-ranking as a text-generation task and explored zero-shot prompting for *generating* an ordered list (Ma, Zhang, et al., 2023; Pradeep et al., 2023; Qin et al., 2023; Sun et al., 2023) or creating the ordered-list by *sorting* the probabilities of generated token (Ma, Zhang, et al., 2023; Zhuang et al., 2023). The long context-window of LLMs facilitates the zero-shot listwise re-ranking approaches, resulting in them outperforming zero-shot pointwise approaches. In pointwise approaches, the model determines relevance for each document individually (returning “True” or “False” tokens), and the probabilities of these tokens are used as relevance scores ((Ma, Zhang, et al., 2023)).

2.4 Query by Example Information Retrieval Systems

There exist many scenarios where it is more convenient for the users to express their information need by providing relevant examples instead of formulating keyword queries. Generally, the example instances can be defined in any form including but not limited to textual documents, user profiles and images (Lissandrini, Mottin, Palpanas, & Velegarakis, 2019). Candidate/talent search at LinkedIn is an example of the **Query-by-Example** (QBE) problem where the searcher query the system by giving one or several ideal candidates for a given position as a query.

Query-By-Document (QBD) is a special case of the QBE problem where the examples are in the form of one or multiple documents. In particular, professional and domain-specific search such as legal case retrieval Abolghasemi, Verberne, and Azzopardi (2022); Althammer, Hofstätter, Sertkan, Verberne, and Hanbury (2022); Askari and Verberne (2021); M.-Y. Kim, Rabelo, and Goebel (2019); Shao et al. (2020), scientific literature retrieval Cohan, Feldman, Beltagy, Downey, and Weld (2020); Mysore, O’Gorman, McCallum, and Zamani (2021), patent retrieval Fujii, Iwayama, and Kando (2007); Piroi and Hanbury (2019) and cross-referencing a news article on a specific topic across sources Y. Yang et al. (2009) are examples of QBD. Williams, Wu, and Giles (2014) propose SimSeerX, a QBD system for retrieving academic documents which combines multiple similarity function and demonstrate it is scalable to a collection of 3.5 million academic documents. Weng et al. (2011) incorporate a two-stage retrieval problem for QBD where in first stage the documents in the collection are encoded into dense vector using dimension reduction techniques and in the second stage locality sensitive hashing is used for quick ranking.

Most of the prior work is focused on one example document as the input and there are a few studies where we have multiple example documents. Lissandrini, Mottin, Palpanas, and Velegarakis (2018) investigate a specific case of having multiple examples where the user issues graph-based queries consistent of tuples against a

knowledge graph. El-Arini and Guestrin (2011) introduce an optimization function based on a probabilistic and concept-specific notion of influence between documents to model query documents in scientific publication search domain. D. Zhang and Lee (2009) address the query by multiple examples as a one-class text classification and use support vector machines to tackle it. Zhu and Wu (2014) adopt a two-stage ranking algorithm where they use topic modeling to formulate a keyword query from example documents and retrieve a set of candidate documents. Then, they use PU learning algorithms to re-rank the set of retrieved candidate documents.

Lastly, query by multiple examples can be viewed as relevance feedback (Salton & Buckley, 1990) where the example documents can be regarded as feedback documents from the user. Smucker and Allan (2006) investigate the “find-similar” feature in some commercial search engines as a form of manual feedback from user and study user behavior and its possible effect on retrieval performance.

CHAPTER 3

ENTITIES AS THE CONTEXTUAL INFORMATION SOURCE

Statement of Contribution

The works described in this chapter, namely *Exploring Summary-Expanded Entity Embeddings for Entity Retrieval* (3.1) and *Local and Global Query Expansion for Hierarchical Complex Entity-centric Queries* (3.2) were published in the EYRE workshop at CIKM 2018 (Naseri S. & B., 2018) and ECIR 2019 (Dalton, Naseri, Dietz, & Allan, 2019), respectively. I was the lead author in the former publication, designed the embedding model and carrying out the experiments. I designed and conducted the experiments regarding the global expansion study on the complex entity-centric queries, in the latter publication.

Knowledge cards, conversational answers, and other focused responses to user queries in current search engines are relying on the availability of rich information for named entities and search based on knowledge graphs. In particular, the *entity retrieval* task has been defined as returning a ranked list of entities relevant to the user’s query to directly answer the queries from knowledge bases. This task is typically approached by finding entities with a “meaning” that is similar to the query.

Beyond questions with a focused response like [“Who won the James Beard Award for best new chef 2018?”], there exist questions which require multifaceted essay-like responses such as [“What are the causes of the Civil War?”] that span a rich variety of subtopics with hierarchical structure. These ‘complex’, multifaceted, hierarchical questions can also take advantage of the words and entities related to their “meaning” and leverage more contextual information about the user’s query.

Capturing the semantic similarity between queries and documents, and representing the context beyond a bag-of-words has been a long-standing problem in information retrieval. Researchers have proposed leveraging knowledge bases that offer information about entities and their relationships to enrich contextual understanding (Lee et al., 2015). Furthermore, traditional word embedding methods like Word2Vec (Mikolov, Sutskever, et al., 2013) offer semantic understanding by assigning low-dimensional vectors to terms, with the semantics derived from capturing co-occurrence information between terms using a likelihood approximation of their appearance within a window of text.

In this Chapter, we investigate the following research question: “How can entities, as source of contextual information, enhance the performance of information retrieval systems with entity-focused queries?” To answer this question, we explore two distinct approaches: 1) learning dense embedding representations of entities, and 2) utilizing corpus-based entity information through probabilistic retrieval methods.

3.1 Exploring Summary-Expanded Entity Embeddings for Entity Retrieval

We study the task of entity retrieval where the queries are entity-centric and the output is a ranked list of relevant entities, in contrast to ad-hoc retrieval where the focus is on retrieving ranked documents. We design an entity embedding representation in Section 3.1.1 and hypothesize that mapping the query to the entity space and comparing with the retrieved entities will improve the retrieval results. In Section 3.1.2 we describe our scheme of retrieval and validate our hypothesis in Section 3.1.4.

3.1.1 Summary-Expanded Entity Embeddings

Following Ni et al. (2016), we learn a joint entity-word embedding to capture semantic relationship between words and entities. This approach is built upon the skip-gram model proposed by Mikolov, Sutskever, et al. (2013). For training the entity embedding model, we require a corpus that could provide the necessary context for each entity. To this end, we utilize the internal linking structure in Wikipedia. An internal link in a Wikipedia page consists of both a hyperlink to another Wikipedia article and a surface form that represents the linked article. We preprocessed all Wikipedia pages by replacing the surface form of each internal link with the title of the referenced article.

Consider the following excerpt, where links to other Wikipedia articles (entities) are represented by italics:

Harry Potter is a series of *fantasy novels* written by British author *J. K. Rowling*. The novel chronicle the life of a young *wizard*, *Harry Potter*, and his friends *Hermione Granger* and *Ron Weasley*, all of whom are students at *Hogwarts School of Witchcraft and Wizardry*.

The excerpt will be replaced by:

Harry Potter is a series of `Fantasy_literature` written by British author `J._K._Rowling`. The novels chronicle the life of a young `Magician_(fantasy)`, `Harry_Potter (character)`, and his friends

`Hermione_Granger` and `Ron_Weasley`, all of whom are students at `Hogwarts`.

where the link is replaced by the corresponding article’s title and spaces are replaced by underscores. Now each entity in the original excerpt is considered as a single “term”, and an embedding is learned based on the Skipgram model.

Furthermore, in knowledge graphs like DBpedia, each entity is accompanied by an abstract where human annotators highlight key related entities. The final embedding of a target entity is then calculated by averaging the embedding vectors of these referred entities within the abstract.

Implementation Details. To train the entity embeddings, we use the full article of Wikipedia pages obtained from the DBpedia 2016-10 dump. We take advantage of the Word2Vec implementation in gensim (Řehůřek & Sojka, 2010) version 3.4.0 with parameters as follow: window-size = 10, sub-sampling = 1e-3, and cutoff min-count = 0. The learned embedding dimension is equal to 200 and we learned embeddings of 3.0M entities out of 4.8M entities available in Wikipedia.

3.1.2 General Scheme of Retrieval

Given a query, q , that targets a specific entity, our task is to return a ranked list of entities relevant to the query. In our approach each entity is represented by a short textual description, specifically its short abstract in DBpedia. A list of candidate entities is retrieved using term-based retrieval models such as query likelihood model (Ponte & Croft, 1998), efficiently creating a large pool of candidates.

In our model, we try to enhance the accuracy of entity retrieval by representing queries and entities by their corresponding embedding vectors. We explore two methods to represent query and entity embedding vectors: **WORDVEC** and **ENTITYVEC**.

In the **WORDVEC** model each query is represented by the average of the embedding vector of the query’s terms. Entities are also represented in a similar way, by averaging

Table 3.1: Query and retrieved entity representations for **WORDVEC** and **ENTITYVEC** models.

Model	Query	Retrieved Entity
WORDVEC	Average of query terms' embedding vectors	Average of embedding vectors of terms in the entity's abstract
ENTITYVEC	Average of query entities' embedding vectors	Average of embedding vectors of related entities in the entity's abstract

over the embedding vectors of the terms in the entity's abstract. The GloVe (Pennington et al., 2014) pre-trained word embedding is used for the word embedding vector in the **WORDVEC** model.

In the **ENTITYVEC** model, we utilize the entity representation introduced in Section 3.1.1 and represent queries by the average of the embedding vectors of their entities. The entities in the query are annotated using the TagMe (Ferragina & Scaiella, 2012) mention detection tool.

For both **WORDVEC** and **ENTITYVEC** the similarity between query and the document is calculated by the cosine similarity between their respective embedding vectors. The final entity retrieval score is obtained by linear interpolation of the baseline, **WORDVEC**, and **ENTITYVEC** models. We summarize the final embedding vector for query and entity in table 3.1.

3.1.3 Experimental Setup

Data set. Our experiments are conducted on the entity search test collection DBpedia-Entity v2 (Hasibi et al., 2017). This dataset originally consists of queries gathered from the seven previous competitions with relevance judgment on entities from DBpedia version 2015-10. For word embeddings, we use the GloVe (Pennington et al., 2014) pre-trained word embedding with 300 dimensions. The word embeddings were extracted from a 6 billion token collection (the Wikipedia dump 2014 plus Gigaword 5). The implementation details of entity embeddings are described in Section 3.1.1.

Data Processing. Retrieval results were obtained using the index built from the abstract of the entities. We use TagMe (Ferragina & Scaiella, 2012) as the mention detection tool for the entities in the queries.

Evaluation Metrics. Mean Average Precision (MAP) of the top-ranked 1000 entities is selected as the main evaluation metric to evaluate the retrieval effectiveness. Furthermore, we consider precision of the top 10 retrieved entities (P@10). Since we have graded relevance judgment, we also report nDCG@10. Statistically significant differences in performance are determined using the two-tailed paired t-test computed at a 95% confidence level based on the average precision per query.

3.1.4 Results

We explore the results of our entity representation model atop two baselines: the query likelihood language model retrieval (LM) (Ponte & Croft, 1998) and the query expansion approach based on relevance modeling (RM3) (Lavrenko & Croft, 2001).

Table 3.2 shows the results of our model on top of the LM baseline for short and verbose query subsets as well as their union. Both proposed methods outperform the baseline LM model, suggesting that there is value in both our **ENTITYVEC** and **WORDVEC** representation. Combining the two methods yields even greater accuracy across all measures, indicating that the two methods are complementary. We observe both **WORDVEC** and **ENTITYVEC** improve verbose queries (queries longer than four terms) more than short queries (particularly measured by MAP). We speculate the reason is, the additional terms in the verbose queries disambiguate the user’s information need, thus it better matches with the relevant entities.

Since some of the queries don’t have entity mentions, the number of ties in the linear combination of LM and **ENTITYVEC** with the LM approach is more than the number of ties of the linear combination of LM and **WORDVEC** with LM.

Table 3.2: Effect of **WORDVEC** and **ENTITYVEC** models on top of LM baseline for verbose, short queries and their union. Superscripts [†], [‡], and [§] indicate statistical significance over the LM, LM+**WORDVEC**, and LM+**ENTITYVEC**, respectively.

Verbose Queries							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
LM	0.1609	-	0.1992	-	0.2261	-	-
LM + WordVec	0.1708 [†]	+6.15%	0.2168 [†]	+8.84%	0.2429 [†]	+7.43%	171/14/77
LM + EntityVec	0.1731 [†]	+7.58%	0.2218 [†]	+11.35%	0.2415 [†]	+6.81%	162/28/72
LM + WordVec + EntityVec	0.1786^{†‡}	+11%	0.2328^{†‡§}	+16.87%	0.2554^{†‡§}	+12.96%	189/16/57
Short Queries							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
LM	0.2445	-	0.2922	-	0.3357	-	-
LM + WordVec	0.2498	+2.17%	0.2956	+1.16%	0.3417	+1.79%	111/23/71
LM + EntityVec	0.2532	+3.56%	0.2985	+2.16%	0.3454	+2.89%	92/49/64
LM + Wordvec + EntityVec	0.2635^{†‡§}	+7.77%	0.3034^{†‡}	+3.83%	0.3531^{†‡}	+5.18%	135/20/50
All Queries							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
LM	0.1976	-	0.2400	-	0.2742	-	-
LM + WordVec	0.2055 [†]	+3.99%	0.2514 [†]	+4.75%	0.2863 [†]	4.41%	282/37/148
LM + EntityVec	0.2083 [†]	+5.41%	0.2555 [†]	+6.45%	0.2871 [†]	+4.70%	254/77/136
LM + WordVec + EntityVec	0.2159^{†‡§}	+9.26%	0.2638^{†‡§}	+9.91%	0.2983^{†‡§}	+8.78%	324/36/107

We investigate the effect of our entity vector models on different types of queries available in our dataset in Table 3.3. Since the queries are of such diverse types, it is not surprising to observe some variation. We see that the **WORDVEC** model does not show a significant improvement in the results of SemSearch-ES, ambiguous keyword queries, and QALD-2, natural language question-answering queries. Since SemSearch-ES queries are mostly ambiguous keyword queries, it is possible that the **WORDVEC** representations are not specific enough to be helpful. Finally, we evaluate the proposed methods in the pseudo-relevance feedback scenario, utilizing RM3 as the state-of-the-art PRF method that has been shown to perform well in various collections (Lv & Zhai, 2009). Similar to LM baseline, in Table 3.4 we observe **WORDVEC** and **ENTITYVEC** improve over the RM3 which means our embedding-based methods are complementary to the keyword-matching expansion approaches. Performing query expansion, more specifically RM3 method here, is well-known to improve retrieval performance. However, in this dataset RM3 results are substantially worse than the simple LM showed in Table 3.3

Table 3.3: Effect of **WORDVEC** and **ENTITYVEC** models on top of LM baseline for different query types. Superscripts [†], [‡], and [§] indicate statistical significance over the LM, LM+**WORDVEC**, and LM+**ENTITYVEC**, respectively.

SemSearch-ES							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
LM	0.3188	-	0.2805	-	0.3901	-	-
LM + WordVec	0.3242	+1.69%	0.2726	-2.82%	0.3908	+0.18%	42/27/44
LM + EntityVec	0.3365 [†]	+5.55%	0.2832 [‡]	+0.96%	0.4014	+2.9%	45/45/23
LM + WordVec + EntityVec	0.3358	+5.33%	0.2867 [‡]	+2.21%	0.3995	+2.41%	57/15/41
ListSearch							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
LM	0.1683	-	0.2800	-	0.2431	-	-
LM + WordVec	0.1724	+2.44%	0.2878	+ 2.79%	0.2493	+2.55%	58/11/46
LM + EntityVec	0.1854 ^{†‡}	+10.16%	0.2957	+5.61%	0.2597 [†]	+6.83%	75/8/32
LM + WordVec + EntityVec	0.1874 ^{†‡}	+11.35%	0.2991 [†]	+6.82%	0.2673 ^{†‡}	+9.95%	76/5/34
INEX-LD							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
LM	0.1593	-	0.2596	-	0.2800	-	-
LM + WordVec	0.1619	+1.63%	0.2747	+5.82%	0.2908	+3.86%	54/5/40
LM + EntityVec	0.1788 ^{†‡}	+7.85%	0.2859 [†]	+10.13%	0.3077 [†]	+9.89%	62/9/28
LM + WordVec + EntityVec	0.1837 ^{†‡§}	+15.32%	0.2949 ^{†‡}	+13.6%	0.3201 ^{†‡§}	+14.32%	71/5/23
QALD-2							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
LM	0.1554	-	0.1907	-	0.2224	-	-
LM + WordVec	0.1557	+0.19%	0.1929	+1.15%	0.2226	+0.09%	62/30/48
LM + EntityVec	0.1653 ^{†‡}	+ 6.17%	0.2100 ^{†‡}	+10.12%	0.2338	+5.13%	94/18/28
LM + WordVec + EntityVec	0.1653 ^{†‡}	+6.17%	0.2100 ^{†‡}	+10.12%	0.2338	+5.13%	94/18/28

Table 3.4: Effect of **WORDVEC** and **ENTITYVEC** models on top of RM3 baseline for verbose, short queries and their union. Superscripts [†], [‡], and [§] indicate statistical significance over the RM3, RM3+**WORDVEC**, and RM3+**ENTITYVEC**, respectively.

Verbose Queries							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
RM3	0.1614	-	0.2103	-	0.2264	-	-
RM3 + WordVec	0.1714 [†]	+6.2%	0.2286 [†]	+8.7%	0.2459 [†]	+8.61%	166/13/83
RM3 + EntityVec	0.1759 [†]	+8.98%	0.2233 [†]	+6.18%	0.2435 [†]	+7.55%	167/31/64
RM3 + WordVec + EntityVec	0.1810 ^{†‡§}	+12.14%	0.2298 ^{†§}	+9.27%	0.2508 ^{†§}	+10.78%	185/15/62
Short Queries							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
RM3	0.2387	-	0.2902	-	0.3289	-	-
RM3 + WordVec	0.2465	+3.27%	0.2941	+1.34%	0.3369	+2.43%	117/19/69
RM3 + EntityVec	0.2524 [†]	+5.74%	0.2976	+2.55%	0.3397 [†]	+3.28%	104/51/50
RM3 + WordVec + EntityVec	0.2546 ^{†‡}	+6.66%	0.3010 ^{†‡}	+3.72%	0.3461 ^{†‡}	+5.23%	131/15/59
All Queries							
Method	MAP@1000		P@10		nDCG@10		Win/Tie/Loss
RM3	0.1954	-	0.2454	-	0.2714	-	-
RM3 + WordVec	0.2044 [†]	+4.60%	0.2574 [†]	+4.88%	0.2859 [†]	+5.34%	283/32/152
RM3 + EntityVec	0.2095 [†]	+7.21%	0.2559 [†]	+4.27%	0.2857 [†]	+5.26%	271/82/114
RM3 + WordVec + EntityVec	0.2133 ^{†‡§}	+9.16%	0.2610 ^{†§}	+6.35%	0.2926 ^{†‡§}	+7.81%	316/30/121

3.2 Local and Global Query Expansion for Hierarchical Complex Entity-centric Queries

A complex topic (i.e. query) T consists of heading nodes constructed in a hierarchical topic tree; an example is shown in Figure 3.1. Each heading node, h , represents the subtopic elements. For example, a complex topic with subtopics delimited by a slash would be: “Urban sprawl/Effects/Increased infrastructure and transportation cost”. This consists of three heading nodes - the leaf heading is “Increased infrastructure and transportation cost” with the root heading “Urban sprawl” and intermediate heading “Effects”. The tree structure provides information about the hierarchical relationship between subtopics. In particular, the most important relationship is that the root heading is the main focus of the overall topic.

Given a complex topic tree T , the outline consists of a representation for each of the subtopic heading nodes $h \in H$. At the basic level, each heading contains its word representation from text, $W : \{w_1, \dots, w_k\}$, a sequence of words in the subtopic. Beyond words, each heading can also be represented by features extracted by information extraction and natural language processing techniques, for example part of speech tags and simple dependence relationships.

In particular, we hypothesize that another key element of effective retrieval for complex topics requires going beyond words to include entities and entity relationships. Therefore, we propose representing the topic as well as documents with entity mentions, T_M and D_M respectively, where each has $M : \{m_1, \dots, m_k\}$ with m_k a mention of an entity e in a knowledge base. Given an entity-centric corpus and task along with rich structure, the mix of word and entity representation offers significant potential for retrieval with complex topics. The result is sequence of ordered entities within a heading with provenance connecting the entity annotations to free text. In TREC CAR as well as adhoc document retrieval, this representation is (partially) latent - it must be inferred from the topic text.

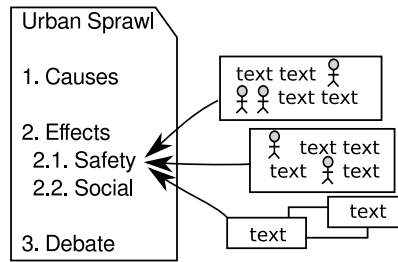


Figure 3.1: Example of a complex topic from the TREC Complex Answer Retrieval track.

3.2.1 Topic Expansion Model

We study two expansion approaches over diverse types of representations, based on words and entities: 1) expansion based on local query-specific relevance feedback and 2) expansion based on global word-entity embedding similarity. To specify the representations we use different term vocabularies, $v \in V$, for example:

- Words, $W : \{w_1, \dots, w_k\}$ are unigram words from the collection vocabulary.
- Entities, $E : \{e_1, \dots, e_k\}$ are entities from a knowledge base, matched based on their entity identifiers.

Note that entities may have multiple vocabularies that interact with one another. We can match entities to word representations using the entity names and aliases $A : \{a_1, \dots, a_k\}$ derived from their Wikipedia name, anchor text, redirects, and disambiguation pages.

To perform effective expansion, our goal is to estimate the probability of relevance for an entry in the vocabulary with respect to the complex topic, T . In other words, regardless of the underlying expansion method, the overarching goal is to identify the latent representation of the topic across all vocabulary dimensions: $p(V|T)$. However, a single expansion model for an entire complex topic is unlikely to be effective. For both expansion methods we also build a mixture of fine-grained expansions for each

subtopic node that are combined. For every type in the vocabulary V , and for every heading node $h \in H$, we create a feature, $f(h, D)$.

In Table 3.5 we illustrate different approaches for expansion that include three dimensions of the expansion: the expansion method, the representation type, and which subtopic to expand. An example is, [Antibiotic use in livestock/Use in different livestock/In swine production]. In this case, $R = [\text{Antibiotic use in livestock}]$ is the root, $I = [\text{Use in different livestock}]$ is an intermediate node, and $H = [\text{In swine production}]$ is the leaf heading. We vary the topic representation using differing combinations of these three elements. The most common approach by participants in TREC CAR is to simply concatenate the RIH context into one query and to ignore the heading relationships or boundaries. In contrast, our fine-grained method preserves these elements and handles them separately.

Features are combined using a log-linear model with parameters, θ . The number of these features is limited to approximately 10. This scale allows it to be learned efficiently using coordinate ascent to directly optimize the target retrieval metric. All of the score-level features, both heading derived and feedback, correspond to queries that can be expressed natively in the first pass matching phase of a search system.

Relevance Model Expansion. The basic building block is the Relevance Model formulation to infer latent words or entities from pseudo-relevant feedback documents (Lavrenko & Croft, 2001). We derive a distribution over all types of the vocabulary. In this case, $p(D = d|T)$ is the relevance of the document to the topic, derived from score for the document under the query model. The $p(V|d)$ is the probability of the vocabulary under the language model of the document using that representation.

Embedding-based Expansion. We utilize the entity embedding representation introduced in Section 3.1.1 to represent entities and then compute embedding-based similarity for both explicit entity mentions as well as words, two types from the vocabulary. For the global similarity between dense embedding vectors we use the

Table 3.5: Examples of topic expansion features across word and entity vocabularies. All features are for R , I , and H nodes separately. The example topic is: [Antibiotic use in livestock/Use in different livestock/In swine production]. The entities identified in the topic are: [Antibiotics, Livestock/ Livestock/ Domestic pig, Pig farming]

Name	Description	Feature Example
RIH-QL	Representing words from the root, intermediate, and leaf subtopics	(antibiotic use livestock different swine production)
RIH-IDs-Embed	Representing expanded entities from global embeddings from the root, intermediate, and leaf subtopics using their IDs	Antibiotics \rightarrow Tetracycline.id Livestock \rightarrow Cattle.id Pig farming \rightarrow (Animal husbandry).id
H-Names-Embed	Expansion of entity names within the leaf subtopic using global embeddings	Pig farming \rightarrow (animal husbandry dairy farming poultry ubre blanca)
R-Aliases-Embed	Expansion of aliases of entity within the root subtopic using global embeddings	Tetracycline \rightarrow (tetracyclin sumycin hydrochloride) Cattle \rightarrow (cow bull calf bovine heifer steer moo)

cosine similarity. In addition to expanding each subtopic node individually, we also perform expansion of the complete topic tree as a whole. The embedding vector of a node (or entire query tree) is represented as the average (mean) of the embedding vector of each element within it.

3.2.2 Experimental Setup

Data. The primary dataset used for experiments is from the TREC Complex Answer Retrieval (CAR) track, v2.1 (Dietz et al., 2018), released for the 2018 TREC evaluation. We explained the CAR dataset in Section 2.2.1.4 in detail. As a brief reminder, each topic consists of a Wikipedia article’s title, heading, and subheading. For evaluation, we take advantage of the “Tree Qrela” automatic judgments. These judgments consider intermediate paragraph headings, thus containing more relevance

judgments than the older CAR “Hierarchical Qrels”. This experimental protocol follows the Y2 and Y3 task definitions, performed on the Y1 query data because automatic judgments are only available on this set. We use the standard V2 of the paragraph collection for the retrieval. It consists of approximately 30 million paragraphs from Wikipedia from December 2016.

Knowledge base. We use the non-benchmark articles from Wikipedia as a knowledge base. These include the full article text, including the heading structure. It does not include the infobox and other data that was excluded in the CAR pre-processing. In addition to the text, we use anchor text, redirects, and disambiguation metadata derived from the article collection and provided in the data.

Evaluation measures. We use the standard measures reported in TREC CAR evaluations. The primary evaluation measure is Mean Average Precision (MAP). We report R-Precision, because the number of relevant documents in TREC CAR varies widely across topics. The NDCG@1000 metric is included following standard practice in the track. For statistical significance, we use a paired t-test and report significance at the 95% confidence interval.

System Details. The TREC CAR paragraph collection is indexed using the Galago ¹ retrieval system, an open-source research system. The query models and feedback expansion models are all implemented using the Galago query language. The paragraphs are indexed with the link fields to allow exact and partial matches of entity links in the paragraphs. Stopword removal is performed on the heading queries using the 418 INQUERY stopword list. Stemming is performed using the built-in Krovetz stemmer. In our score fusion model we use a log-linear model combination of different features for ranking. The model parameters, θ are optimized using coordinate ascent to directly optimize the target retrieval measure, Mean Average precision (MAP).

¹<http://www.lemurproject.org/galago.php>

Table 3.6: Text-based baselines and expansion methods. * indicates significance over the RH-SDM run.

Model	MAP	R-Prec	NDCG
RIH-QL	0.110	0.088	0.228
RH-SDM	0.132	0.109	0.248
RH-SDM-RM3	0.127	0.102	0.243
L2R-SDM-RM3	0.142*	0.107	0.257*
Embedding-Term	0.143*	0.119*	0.261*
GUIR (neural)	0.137	0.112	0.237
GUIR-Exp (neural)	0.142*	0.117	0.242

Table 3.7: Learned feature weights of combination of SDM and RM3 over different outline levels using L2R.

Model	Weight
RIH-QL	0.288
R-SDM	0.153
H-SDM	0.340
RH-SDM	0.108
RH-SDM-RM3	0.110

The implementation of the model is available in the open-source RankLib learning-to-rank library. The implementation details of the entity embeddings is described in Section 3.1.1.

Query Entity Annotation. The topics in TREC CAR do not have explicit entity links. To support matching paragraph entity documents, we annotate the complex topic headings with entities. Entity linking is performed on each heading for both the train and test benchmark collections. We use the open-source SMAPH² entity linker, a state-of-the-art approach at the time of our experiments .

Document Entity Annotation. For entity mentions in documents we use the existing entity links provided in Wikipedia. We note that the entity links in Wikipedia are sparse and biased. By convention only the first mention of an entity in an article is annotated with a link. This biases retrieval based on entity identifiers towards paragraphs that occur early in a Wikipedia article.

3.2.3 Results

Word-based Retrieval and Expansion. We first evaluate standard text retrieval methods for heading retrieval. The results are shown in Table 3.6. The baseline model, RIH-QL, is a standard bag-of-words query-likelihood model (Ponte & Croft, 1998) on all terms in the topic. All other runs show statistically significant gains over this simple baseline. The table also shows results for the Sequential Dependence Model (SDM) that uses the root and leaf subtopics of the heading. We also experimented with other variations (H-QL, RIH-SDM, RH-QL, etc), but these are all outperformed by RH-SDM. RH-SDM was the best performing unsupervised model for this collection in TREC 2018. We also evaluate using a relevance model term-based expansion on top of the best SDM run. We find that the RM3 performance is insignificantly worse than the SDM baseline, demonstrating the PRF based on words is challenging in this environment. We attribute this to the sparseness of relevant paragraphs to the topics, an average of 4.3 paragraphs per topic, with baselines retrieving on average about half of those, 2.2.

We experimented with combining the baseline systems with additional fine-grained SDM components from each part of the query (subtopic) separately and weighting and combining them into a linear model, the L2R-SDM-RM3 method. The features and learned weights are given in Table 3.7. We observe that the H-SDM feature is the most important, putting greater emphasis on the leaf subtopic (approximately 2x the root topic). Combining these baseline retrieval and subtopic heading components results in significant gains over all the models individually, including RH-SDM. The Embedding-Term method is the L2R-SDM-RM3 with addition of global word expansion. The results show a small, but insignificant improvement to the model effectiveness.

²<https://github.com/marcocor/smaph>

Table 3.8: Entity-based expansion with varying latent entity models. * indicates significance over the L2R-SDM-RM3 Baseline.

Model	MAP	R-Prec	NDCG
L2R-SDM-RM3	0.142	0.107	0.257
Entity_Embedding	0.154	0.127	0.277
Entity_Retrieval	0.160*	0.133	0.284*
Entity_Collection_PRF	0.172*	0.146*	0.297*

The bottom of Table 3.6 shows a comparison with one of the leading neural ranking models (at the time that this work is done) from the Georgetown University IR group (GUIR). It uses the PACRR neural ranking architecture modified with heading independence and heading frequency context vectors (MacAvaney et al., 2018). The second row (Exp) adds expansion words of the topic’s query terms. Interestingly, the learned GUIR neural run does not improve significantly over the RH-SDM baseline, the SDM model even slightly outperforms it on NDCG. The learned word-based expansion methods L2R-SDM-RM3 and Embedding-Term are both statistically significant over the GUIR base run for MAP, but not statistically significantly different from the Exp run. This indicates that our methods are comparable to state-of-the-art word-based expansion models using deep learning for this collection.

Entity Expansion. We study combining the previous word-based representations with entity representations, and use entities annotated in the query as well as inferred entities from local and global sources: global embeddings, local entity retrieval, and local pseudo-relevance feedback on the paragraph collection. Each of the entity expansion models is a learned combination of subtopic expansions across the different entity vocabularies (identifiers, names, aliases, and unigram entity language models). The results in Table 3.8 show that adding entity-based features improves effectiveness consistently across all entity inference methods. There are benefits to using global entity embeddings, but they are not significant over the baseline. The

local retrieval and collection PRF expansion models both result in significant improvements over the baseline. In particular, the collection entity representation shows the largest effectiveness gains. Additionally, all of the entity-based expansion methods show statistically significant improvements over the GUIR-Exp word-based expansion run. When compared with the baseline word model entity-expansion methods have a win-loss ratio varying from 2.6 up to 4.6. The best method based on collection feedback has 281 losses, 1300 wins, with a win-loss ratio of 4.6. In contrast, the win-loss ratio for the GUIR-Exp model is 1.1, hurting almost as many queries as it helps.

Consequently, we conclude that entity-based expansion methods more consistently improve effectiveness for complex topics when compared with word-based expansion methods.

3.3 Discussion

The work in this chapter was carried out from 2017 to 2019. In this section, we briefly review subsequent research on retrieval tasks involving entity-centric queries and documents. These works primarily build upon advancements made after our initial research, particularly focusing on the application of Transformer models.

Following our exploration of entity-based embeddings for entity retrieval, Gerritse et al. (2020) investigated using Wikipedia2vec (Yamada et al., 2018) embeddings for representing both queries and documents in this task. Their approach involved using the similarity between query and document vectors derived from Wikipedia2vec as a re-ranking step, subsequent to an initial ranking obtained by a fielded probabilistic retrieval model on the DBPedia-Entity V2 dataset. In a subsequent study, with the advent of Transformer architectures, Gerritse et al. (2022) combined E-BERT, an entity-enhanced BERT model, with the monoBERT cross-encoder re-ranking architecture for entity retrieval. This approach achieved state-of-the-art results for entity-

centric queries of DBPedia-Entity V2 dataset, surpassing their previous Word2Vec-based re-ranking strategy.

Further, the mono-BERT (Nogueira & Cho, 2019) re-ranking architecture achieved state-of-the-art performance in TREC-CAR complex entity centric queries achieving an improvement of 2X compared to non-neural and earlier neural baselines.

3.4 Summary

In this chapter, we studied the task of Entity Retrieval that involves retrieving a ranked list of entities for a specific query. We employ the *contextual information* latent in entities’ characteristics such as their relationship with other entities to represent the queries and documents. We developed an entity embedding model that leverages relationships within an entity’s summary stemming from other entities mentioned in the summary for enhanced representation. We applied this model to an entity ranking task, representing both queries and documents within the model. Combining this embedding-based ranking with a traditional Language Model retrieval approach yielded significantly improved performance compared to term-based retrieval alone. Furthermore, our entity-based embedding ranking outperformed a word-based embedding model. Finally, we created a fusion retrieval model integrating the term-based Language Model, word-based embedding ranking, and our entity-based embedding ranking, achieving the most robust results (*Contribution 1.1*).

Further, we developed an entity-aware query expansion method for passage retrieval, applicable to complex, multifaceted, hierarchical queries. Our approach combines both probabilistic retrieval techniques and entity embedding vectors. This allows us to incorporate entities from ‘local’ (corpus-specific entities index) and ‘global’ (general knowledge entities obtained from embedding) sources. The resulting expansion model outperforms the learned combination of probabilistic word-based models by 21%, demonstrating the value of entity-based representations (*Contribution 1.2*).

However, it's important to note that later works using BERT-based models have reported even higher performance levels (Nogueira & Cho, 2019).

CHAPTER 4

PSEUDO RELEVANT FEEDBACK DOCUMENTS AS CONTEXTUAL INFORMATION SOURCE

Statement of Contribution

The works described in this chapter, namely *CEQE: Contextualized Embeddings for Query Expansion* and *CEQE to SQET: A study of contextualized embeddings for query expansion* were published in the ECIR 2021 (Naseri, Dalton, Yates, & Allan, 2021) and Information Retrieval Journal (Naseri, Dalton, Yates, & Allan, 2022), respectively. The latter work is an extension of the former. I was the lead author in both of the publications, and designed the expansion models and the experiments.

Traditional text processing methods often relied on either high-dimensional word-based representations or continuous low-dimensional vectors. One example of the former is Term Frequency-Inverse Document Frequency (TF-IDF) which determines how important a word is to a document within a larger collection. It considers how often the word appears in the document (term frequency) while down-weighting words that are common across many documents (inverse document frequency). On the other hand, continuous low-dimensional vectors were popularized by models such as Word2Vec (Mikolov, Sutskever, et al., 2013). Word2Vec uses neural networks to embed words into a dense vector space which learns term representation by predicting a word based on its context within a sentence, thereby capturing semantic and syntactic word relationships. However, both approaches fundamentally assign context-independent, static representations to words.

Transformer models (Vaswani et al., 2017) leverage attention mechanisms to dynamically understand the context of words within text and address the limitation of static representation. Pre-training on massive amounts of data provides these models with a rich foundation of language knowledge, further enhancing their contextual understanding. This makes them significantly more powerful than previous static models in various NLP and retrieval tasks.

Further, in information retrieval systems, Pseudo-Relevant Documents (PRF) – which are documents initially retrieved in response to the original query and assumed to be relevant – have been extensively studied to provide more context and facilitate query reformulation (query expansion). However, previous work primarily relied on static term representations, such as term-frequency-based methods and static embedding methods.

With the advent of Transformer-based language models, which represent terms according to their surrounding context, in this chapter we investigate the central question of “How can we effectively leverage the context within PRF documents by

utilizing these models for improved query expansion?”. This research addresses a fundamental problem in information retrieval where the core matching algorithms often fail to identify many relevant results in the first candidate pool. To improve retrieval outcomes, we need more effective core matching algorithms that boost Recall, providing a richer foundation for neural re-ranking methods.

4.1 Unsupervised Query Expansion with Transformers

4.1.1 Word and WordPiece representations

In contextualized models, to address the problem of out-of-vocabulary terms, sub-word representation such as WordPieces (Schuster & Nakajima, 2012) is used. For backward compatibility with existing word-based retrieval systems (as well as comparison with previous methods) we use words as the matching unit. We first aggregate WordPiece tokens into a contextualized vector for words. We compute the average embedding vector of word w by $\vec{w} \triangleq \frac{1}{|w|} \sum_{p_i \in w} \vec{p}_i$, where p_i is a WordPiece of word w and $|w|$ is the number of WordPieces in the word w .

4.1.2 Contextualized Embeddings for Query Expansion (CEQE)

We propose the CEQE model that follows in the vein of principled probabilistic language modeling approaches, such as the Relevance Model formulation of pseudo-relevance feedback (Lavrenko & Croft, 2001). In contrast to these approaches that are based on static lexical matching, we formulate relevance based on contextualized vector representations. We build the contextualized feedback model based upon the core Relevance Model (RM) formulation:

$$p(w|\theta_R) \propto \sum_{D \in R} p(w, Q, D) \quad (4.1)$$

where θ_R and R respectively denote the feedback language model and the set of pseudo-relevant documents, i.e., the top retrieved documents, and w , Q and D rep-

resent word, query and document respectively. In the original RM formulation, the joint probability of $p(Q, w, D)$ is broken down as follows:

$$\sum_{D \in R} p(w, Q, D) = \sum_{D \in R} p(w, Q|D)p(D) \quad (4.2)$$

$$= \sum_{D \in R} p(w|Q, D)p(Q|D)p(D) \quad (4.3)$$

$$= \sum_{D \in R} p(w|D)p(Q|D)p(D) \quad (4.4)$$

where Equation 4.4 is derived from the simplifying independence assumption between the query Q and term w . This assumption results in a static representation based on simple word counts and ignores the query explicitly (by assuming that the expansion term w is conditionally independent of Q given D). It only incorporates evidence indirectly through $P(Q|D)$. In contrast, the proposed CEQE parameterization doesn't assume the term w is independent of the query Q and explicitly incorporates the query focus based on similarity with contextualized vector representations. More formally:

$$\sum_{D \in R} p(w, Q, D) = \sum_{D \in R} p(w|Q, D)p(Q|D)p(D) \quad (4.5)$$

which is the same as Equation 4.3.

With a contextualized model it is no longer possible to simply count document terms – they must be grouped, simplified, or compared against a query representation. We explicitly incorporate contextualized query similarity for each word occurrence. We now break down each of the elements in Equation 4.5 in more detail. Following common practice, we assume a uniform probability for $p(D)$. $p(Q|D)$ is the posterior probability of the query given a document from the retrieval model. The retrieval model can be either a Language model with Dirichlet smoothing or even BM25. For BM25 the retrieval scores are mapped to a probability distribution by applying the

Softmax function on the document scores. We propose several methods to calculate $p(w|Q, D)$ below.

Centroid Representation In this approach, we create a model of the whole query and then compare it to the contextualized representation of each word mention (occurrence), m_w . In the centroid representation we define $\sigma(Q)$, the aggregation of all WordPieces of the query. Note that a representation of a query also includes special delimiter tokens. For example, in BERT this would include [CLS] and [SEP] tokens that we find carry contextual importance. We include the [CLS] token in particular because it is often used as a representation of the input with respect to the target task. For the query centroid representation we define σ as the mean of its individual component contextual vectors: we represent query $\sigma(Q)$ by $\vec{Q} \triangleq \frac{1}{|Q|} \sum_{q_i \in Q} \vec{q}$, where q_i is a WordPiece token and $|Q|$ is the length of the query in WordPiece tokens.

We then define $p(w|Q, D)$ by comparing the similarity of individual word mentions to the query centroid representation based on a similarity function δ (e.g., cosine). If m_w^D is a mention of word w in a document D and M_w^D is the complete set of mentions of w :

$$p(w|Q, D) \triangleq \frac{\sum_{m_w^D \in M_w^D} \delta(\vec{Q}, \vec{m}_w^D)}{\sum_{m^D \in M_*^D} \delta(\vec{Q}, \vec{m}^D)} \quad (4.6)$$

The denominator is a normalization constant that considers all word mentions across the entire document to form a probability. This approach is novel because the contextualized vector m_w^D will be different for every occurrence in D because the context surrounding each mention of word w varies.

Term-based Representation In this section we propose an alternative parameterization for $p(w|Q, D)$. Instead of using the centroid of the query to compute a term’s similarity to the entire query, we compute the similarity for each query term separately. If q is a query term and \vec{q} is its corresponding contextualized embedding

vector, this can be formulated as:

$$p(w|q, D) \triangleq \frac{\sum_{m_w^D \in M_w^D} \delta(\vec{q}, \vec{m}_w^D)}{\sum_{m^D \in M^D} \delta(\vec{q}, \vec{m}^D)} \quad (4.7)$$

To select a term for expansion for the query overall we perform an extra step of pooling across the similarities of individual words. This step combines the contextualized word vectors. Function f calculates the semantic similarity of word w with the whole query by combining the semantic similarity of it with each query term q . We define $f_{\max}(w, Q, D) = \max_{q \in Q} p(w|q, D)$ and $f_{\text{prod}}(w, Q, D) = \prod_{q \in Q} p(w|q, D)$ as MaxPool and MulPool, respectively. If Z' is a normalization factor that is the sum over the terms in document D , which is less computationally expensive than summing over all vocabulary terms, these can be defined as:

$$p(w|Q, D) \triangleq \frac{f_{\max/\text{prod}}(w, Q, D)}{Z'} \quad (4.8)$$

The final result of all of these methods is a relevance distribution over terms derived from the contextualized representations in top retrieved documents. The result is an updated query language model that can be used on its own or combined with other representations. In our experiments, we follow the *standard* variant (Abdul-Jaleel et al., 2004) of Relevance Model, RM3 which is designed to maintain information in the original query model as well as the information gained from the behaviour of the returned documents by linearly interpolating the relevance model with the original query model:

$$P'(w|\theta_R) = \lambda P(w|\theta_R) + (1 - \lambda)P(w|Q) \quad (4.9)$$

where $P(w|Q)$ is the original query language model which without loss of generality we confine our experiments to Query Likelihood (QL).

4.2 Supervised Query Expansion with Transformers (SQET)

We propose SQET, which models the problem of query expansion as a **classification task** to classify the expansion term as relevant or non-relevant to the query. Given a query Q and an expansion term w , either with a context or without, a BERT-based encoder computes the relevance score between the query Q and the expansion term w . Note that SQET is a discriminative (classification) model that learns the boundary between the relevant and non-relevant classes, rather than a generative model which learns a distribution of the individual classes. We build the set of candidate expansion terms based on the pseudo-relevance documents retrieved using a traditional IR model.

SQET represents a model that computes the relevancy score between the query Q and the expansion term w without any context for w . Following the same notation as Devlin et al. (2019) we feed the query as sentence A and the expansion term as sentence B: $[\text{CLS}], Q, [\text{SEP}], w, [\text{SEP}]$. We feed the final hidden state corresponding to $[\text{CLS}]$ in the model to a single layer neural network with softmax activation function which outputs the probability that the term w is relevant to the query Q . This produces a query expansion term probability distribution over the vocabulary. Following the standard variant of the Relevance Model, RM3 (Equation 4.9) we perform a linear interpolation of the SQET expansion query terms with the Query Likelihood score of the original query.

SQET-Context aims to leverage the contextual information of the candidate expansion term in the retrieved pseudo-relevant documents. As mentioned earlier the pool of candidate expansion terms is created from the pseudo-relevant documents' terms. To provide the terms with context we define a fixed window of terms around the candidate expansion term's mention in the pseudo-relevant document with size c . Unlike the model from Dai and Callan (2019), BERT-MaxP, that calculates the relevance score of the documents' passages to the query and re-ranks the result based

on the calculated score, we calculate the relevancy of each *term* of the pseudo-relevant document to the query in order to improve the first round of retrieval by expanding the query with top relevant terms. Since there could be multiple mentions of a candidate expansion term in the pseudo-relevant document, we define the context of the i th mention of the candidate expansion term w as $\text{context}(m_w^i)$.

We form the input of the BERT-based encoder by concatenating the Query Q and the context of the the i th mention of the candidate expansion term $\text{context}(m_w^i)$: $[[\text{CLS}], Q, [\text{SEP}], \text{context}(m_w^i), [\text{SEP}]]$.

Similar to the SQET model, by feeding the [CLS] final hidden state to a feed forward model, we get the probability of $\text{context}(m_w^i)$ being relevant to the query Q . To determine the relevance score of the candidate expansion term w , we apply inference using the following three aggregation functions:

- **Max** represents the probability of the candidate expansion term w by maximum relevancy score between Q and $\text{context}(m_w^i)$.
- **Weighted Sum (wSum)** represent the probability of the candidate expansion term w by the weighted sum of the relevancy score between Q and $\text{context}(m_w^i)$ derived from BERT. The formulation is as follow:

$$p(w) = \frac{1}{Z} \sum_{m_w^i \in M_w} \text{tf}(w, \text{context}(m_w^i)) \times \text{BERT}(Q, \text{context}(m_w^i)) \quad (4.10)$$

where M_w is the set of mentions of the candidate expansion term w , $\text{tf}(w, \text{context}(m_w^i))$ is the frequency of term w in $\text{context}(m_w^i)$, $\text{BERT}(Q, \text{context}(m_w^i))$ is the relevancy score between $\text{context}(m_w^i)$ and Q calculated by a BERT-based ranker, and Z is merely a normalizer allowing for the weights to be turned into a probability distribution.

- **invRank** represents the probability of the candidate expansion term w by aggregating the relevancy score between Q and $\text{context}(m_w^i)$ according to the inversed log of rank of the $\text{context}(m_w^i)$. The formulation is as follow:

$$p(w) = \frac{1}{Z} \sum_{m_w^i \in M_w} \frac{1}{\log_2(\text{rank}(\text{context}(m_w^i)) + 1)} \times \text{BERT}(Q, \text{context}(m_w^i)) \quad (4.11)$$

M_w and $\text{BERT}(Q, \text{context}(m_w^i))$ and Z are defined as mentioned above.

4.3 Experimental Setup

4.3.1 Datasets

We study the models on multiple standard TREC benchmark datasets: Robust, Deep Learning, and Complex Answer Retrieval (CAR). For SQET we focus on its behavior in the well-studied adhoc Robust dataset.

- **Robust** The corpus consists of Tipster disks 4 and 5 containing approximately 528K newswire articles. The evaluation topics are the 250 Robust topics (301-450, 601-700). We use the titles as queries.
- **TREC Deep Learning** The 2019 TREC Deep Learning (TREC19-DL) Track created large labeled datasets for ad-hoc search. We perform the full document ranking task with the goal of testing new expansion methods to improve effectiveness. The evaluation has 43 test queries from Bing, and the corpus consists of 3.2 million web documents. Documents are rated on a four point graded relevance scale. The primary measure is nDCG@10.
- **TREC CAR** TREC Complex Answer Retrieval (CAR) (Dietz et al., 2018) is a dataset curated for the TREC Complex Answer Retrieval track introduced in

2017 to address retrieval for complex topics. We provide more details on the TREC CAR dataset in Section 2.2.1.4.

Evaluation Metrics. Since we focus on introducing relevant documents to a candidate pool for downstream ranking, we consider both Recall-focused metrics (Recall@100, Recall@1000, MAP) as well as precision-based measures (P@10/20, nDCG@10/20). For Robust, in order to compare with previous works we report precision and nDCG at cut-off 20. We report the official primary measure for TREC19-DL, nDCG@10. For significance testing, we use a paired t-test with significance at the 95% confidence interval.

4.3.2 Intrinsic expansion judgments

Beyond direct retrieval, we also assess term selection quality intrinsically. We directly measure the utility of individual expansion terms. Following previous work from Imani et al. (2019), we generate this term utility by performing expansion one word at a time. Retrieval effectiveness assesses whether a term is good (helps retrieval), bad (hurts retrieval), or neutral (has no effect). We pool the top thousand candidate expansion terms from all candidate expansion methods. These are issued to the retrieval system with the original query (each with a default weight of 0.5, the default relevance model expansion weight). This approach follows standard relevance model interpolation practice defined in Equation 4.9, which removes the dependence on the original query length (instead of simply appending a word). We measure improvement based on Recall@1000 with a threshold of 0.001. For Robust this results in approximately 500k candidate terms. For the intrinsic evaluations only queries with at least one positive expansion term are used. This is 181 queries for Robust with 10,068 positive terms.

4.3.3 Baselines

4.3.3.1 Unsupervised: CEQE

We study the behavior of the CEQE model in comparison with standard models from probabilistic language modeling. For the baseline retrieval we use BM25 because it is the most widely used first-pass unsupervised ranker used to generate candidate pools. We compare with two static expansion models (Kuzi et al., 2016) and a proven pseudo-relevance feedback model, the Relevance Model (Lavrenko & Croft, 2001). We use the standard relevance model (RM3 variant) that performs linear interpolation of the RM expansion terms with the original query using the Query Likelihood score.

Static Embeddings For static word embeddings we use GloVe (Pennington et al., 2014) embeddings. The pre-trained 300 dimensional Glove word embeddings are extracted from a 6 billion token collection (Wikipedia dump 2014 plus Gigawords 5). These embeddings are the most effective static embeddings for a variety of tasks, including previous work (Diaz et al., 2016) on query expansion. We use the static embeddings with two variations. The *Static-Embed model* (Kuzi et al., 2016) is a global expansion model using GloVe expansion on the target collection vocabulary. For a fair comparison with CEQE, we additionally consider a *Static-Embed-PRF* variant that has its vocabulary limited to terms appearing in the PRF documents.

4.3.3.2 Supervised: SQET

Similar to the unsupervised model, CEQE, we study the behavior of the SQET variants in comparison with the standard models from probabilistic language modeling: BM25 and RM3.

BM25_{invRank} To validate the effect of the scores obtained by BERT on the expansion terms' ranking in the SQET-Context, we replace the BERT-based score with the BM25 score and use the inverse log rank aggregation approach to calculate the final score.

MASK-QE We replace a query term with a [MASK] token in order to see what terms can be in the position of the masked query term. We take advantage of a pre-trained BERT model to predict the masked query term.

4.3.4 System Details

All collections are indexed with the Galago¹ open-source retrieval system for research. The query models and feedback expansion models are all implemented using the Galago query language. We perform stopword removal and stemming using Galago’s stopword list and Krovetz stemmer, respectively.

Contextualized Embedding Model We use BERT because it is the most widely used contextual representation model. We use the pre-trained BERT (BERT-Base, Uncased) model with maximum sequence length of 128. for calculating the contextualized embedding vectors.

- **Unsupervised** Since the documents in Robust are longer than 128 tokens we split the documents into chunks with a maximum size of 128 tokens. For the primary CEQE results in this section we use a single layer of the contextualized representation, the second to last layer (11) of BERT. This layer was shown to be the most effective single layer on NER (Devlin et al., 2019) and it was shown that later layers (before the last) were the most effective word representations for multiple language tasks (Peters, Ruder, & Smith, 2019) that use contextual embeddings as features. Initial preliminary experiments confirmed this finding.
- **Supervised** In SQET-Context, the window size is chosen from {5, 10}. If there are not enough terms surrounding the candidate expansion term, we pad the sentence with [PAD] wordPiece token. For the MASK-QE baseline, we observe that the predicted terms are sensitive to whether the input text is padded. In

¹<http://www.lemurproject.org/galago.php>

order to conduct our experiments with batched input, we set the maximum sequence length of BERT for query input to 12 since the maximum length of tokenized Robust04 queries using WordPiece (Schuster & Nakajima, 2012) is equal to 10.

Neural ranking models For our neural models we adopt CEDR (MacAvaney et al., 2019). In particular, to align with the use of the contextualized models we use the BERT variant. For Robust, we use the CEDR-KNRM model trained by the authors (MacAvaney et al., 2019). Throughout our experiments we refer to the CEDR-KNRM as CEDR. For TREC19-DL we use a CEDR variant trained on a random sample of 1000 MS MARCO train queries with early stopping to terminate when there is no validation improvement for 20 iterations.

Training - Supervised We fine-tune the SQET variants using a TPU v3 with a batch size of 128. The SQET model includes approximately 350K negative and 6.6K positive instances. The Context-SQET model consists of approximately 2M negative and 4K positive instances. To avoid biasing the model towards predicting non-relevant labels, which are approximately 50 times more frequent in the training set, we build each batch by sampling an equal number of relevant and non-relevant expansion terms. For both models, we use Adam (Kingma & Ba, 2014) with the initial learning rate set to 2×10^{-6} , learning rate warmup proportion equal to 0.1 of the training steps, and linear decay of the learning rate.

4.4 Results

4.4.1 Contextualized Query Expansion

We study how to incorporate contextualized embeddings for the task of unsupervised and supervised query expansion. First, we evaluate the retrieval effectiveness of our expansion models in combination with unsupervised retrieval systems, such as BM25. We study this setup because expansion is widely performed on top of these

simple and fast unsupervised baselines. We start with CEQE and baselines on the 2019 Deep Learning Track in Table 4.1 and TREC CAR in Table 4.2. After these, we compare the behavior of CEQE and SQET on the Robust collection.

Deep Learning 19 (Table 4.1) We report the official evaluation measures for the TREC 2019 Deep Learning Track (Craswell, Mitra, Yilmaz, & Campos, 2019) as well as Recall@1000. For nDCG@10, the baseline BM25 retrieval is more effective than all expansion methods. To give an indicator of the BM25 + RM3 parameters, the average parameter settings across the folds is: 15 feedback docs, 85 expansion terms, and interpolation weight of 0.4. We observe that a tuned RM3 outperforms the static embedding methods across all measures. CEQE-MulPool and CEQE-MaxPool also outperform the static embedding model across all measures. The best performing *expansion* method is CEQE-MaxPool, outperforming RM3 (note that this comparison is among the individual expansion methods excluding CEQE-MaxPool-RM3comb, TREC 2019 Median and TREC 2019 Best runs). The interpolation of CEQE-MaxPool and RM3 yields small improvements over MaxPool, indicating that RM3 is not adding significantly new information. We note that given the small sample size (43 topics), none of the unsupervised methods show statistically significant differences between them. As shown later, that requires performing expansion on top of neural rankings.

Although our experimental setup is based on cross-fold validation (rather than tuning on MARCO), we include the reported values from the Deep Learning track overview (Craswell et al., 2019) for reference. Importantly, we observe that the CEQE-MaxPool outperforms all submitted TREC systems on Recall@1000 and is in the top five for Recall@100. Moreover, we observe that the CEQE-MaxPool performs competitively with the TREC 2019 Median run in P@10. It’s noteworthy that the unsupervised CEQE-MaxPool ‘traditional’ model is only slightly lower than the median for P@10 and nDCG@10 with runs that include many state-of-the-art neural models.

Table 4.1: Ranking effectiveness of CEQE on unsupervised baseline retrieval for Deep Learning 2019 Track for the task of full document ranking. The superscript † and ‡ denote statistical significance over BM25 + RM3 and Static-Embed, respectively.

Model	P@10	nDCG@10	mAP@1000	Recall@100	Recall@1000
BM25	0.6535	0.5730	0.3513	0.4053	0.6950
BM25 + RM3	0.6256	0.5343	0.3975 [†]	0.4434 [‡]	0.7750 [‡]
Static-Embed	0.6186	0.5427	0.3373	0.3973	0.7179
Static-Embed-PRF	0.5605	0.4925	0.3166	0.3715	0.6737
CEQE-Centroid	0.5580	0.5580	0.4144 [†]	0.4464 [‡]	0.7804 [‡]
CEQE-MulPool	0.6442	0.5563	0.3724 [†]	0.4295 [‡]	0.7560 [‡]
CEQE-MaxPool	0.6581	0.5614	0.4161 ^{††}	0.4506 [‡]	0.7832 [‡]
CEQE-MaxPool-RM3comb	0.6535	0.5579	0.4178^{††}	0.4507[‡]	0.7843[‡]
TREC 2019 Median	0.6597	0.5834	0.2984	0.3748	0.5484
TREC 2019 Best	0.8093	0.7260	0.4280	0.4670	0.7553

More specifically, the TREC 2019 Median and TREC 2019 Best are among the runs that take advantage of a train dataset with more than 36K queries and are specifically tuned to improve nDCG metric. However, as stated earlier the CEQE models does not take advantage of any train data and its focus is to improve Recall by including relevant documents in top 100 or top 1000 retrieved document to later to be used as a first stage run for the neural re-rankers. Moreover, since CEQE is a query expansion technique it is prone to drift the query by introducing extraneous words (Croft, Metzler, & Strohman, 2010) which can result in drop in the performance in terms of precision-based metrics.

Complex Answer Retrieval (Table 4.2) We follow previous expansion work on CAR (Dalton et al., 2019), and use BenchmarkY1Tree with the root topic titles removed. This is the recommended setup from the CAR organizers, and is an updated version of the widely used hierarchical judgments (and therefore slightly different from reported hierarchical values (Nogueira et al., 2019)). The baselines are comparable to the Lucene runs provided by the track organizers.

The CAR collection is particularly challenging for feedback models because there are few relevant paragraphs per query in the collection, approximately 3.5 on average.

Table 4.2: Ranking effectiveness of CEQE on unsupervised baseline retrieval for the Complex Answer Retrieval (CAR) Track. The superscript † and ‡ denote statistical significance over BM25 + RM3 and Static-Embed-PRF, respectively.

Model	mAP@1000	R-Prec	Recall@100	Recall@1000
BM25	0.1102	0.0857	0.3680	0.5867
BM25 + RM3	0.1119	0.0881	0.3782	0.6056
Static-Embed	0.1144	0.0895	0.3796	0.5900
Static-Embed-PRF	0.1135	0.0879	0.3880 [†]	0.6014
CEQE-Centroid	0.1124	0.0869	0.3806	0.6138 [‡]
CEQE-MulPool	0.1020	0.0801	0.3615	0.6018
CEQE-MaxPool	0.1127	0.0877	0.3801	0.6141 ^{†‡}
CEQE-MaxPool-RM3Comb	0.1122	0.0871	0.3808	0.6155 ^{†‡}

Also, Recall for CAR topics is lower by more than 10% for BM25 and 18% for BM25 + RM3 when compared with the other test collections. The PRF feedback parameters learned on BenchmarkY1Train are 20 feedback paragraphs, 50 feedback terms, and an interpolation weight of 0.9. This indicates almost all weight is being given to the original query (which is also longer with multiple Wikipedia headings).

The results show that the CEQE-MaxPool outperforms the existing static methods for Recall@1000. In fact it provides the only statistically significant improvement over the BM25 baseline. The interpolation of CEQE-MaxPool and RM3 yields marginal improvements over MaxPool alone, indicating the CEQE is relatively robust on its own.

We observe small gains over RM3 from the static embedding models. In particular, the static-embed-PRF has the best Recall@100 of the expansion runs. The static Glove embedding has the best MAP score. We hypothesize that requiring the terms to be in both GloVe and PRF documents is providing a useful filter when there are few relevant documents retrieved. CEQE is competitive and insignificantly different in other measures. All in all, the main objective of CEQE is to do query expansion to include more relevant terms to the query in order to include more relevant documents

in the first stage ranking. Query expansion can introduce query drift due to extraneous words or weighting of terms (Croft et al., 2010). Thus, they perform better with Recall-oriented metrics compared to precision-oriented metrics.

4.4.1.1 Comparing CEQE and SQET on Robust

We now study the behavior of the CEQE unsupervised model and compare it with the SQET supervised model on Robust in Table 4.3. All the Static-Embed variants, CEQE variants and SQET variants outperform the baseline BM25 retrieval method across all measures. MASK-QE is the only expansion method that performs worse than the BM25 baseline.

The static embedding models outperform BM25, but do not perform as well as the Relevance Model (RM3). The Static-Embed-PRF method that only uses terms in the PRF documents’ vocabulary is more effective across all measures over the Static-Embed approach with a global vocabulary. We hypothesize that this may be due to the fact that the query results provide a topically focused vocabulary and filter out generally similar noise. RM3 significantly outperforms the Static-Embed method for MAP, but not other measures. To give an indicator of the BM25 + RM3 parameters, the average parameter settings across the folds is: 22 feedback docs, 71 expansion terms, and interpolation weight of 0.3. We observe that all CEQE variants outperform the static embedding models. The results show CEQE-MaxPool is the best CEQE variant method. The Centroid method is slightly lower than MaxPool, and both outperform multiplicative pooling. The CEQE-MaxPool result outperforms the BM25+RM3 across all measures and in Recall@1000 is significant over both static embedding methods and BM25+RM3, which demonstrates the utility of context-dependent embeddings.

The CEQE-MaxPool-RM3Comb which is a combination of CEQE-MaxPool and RM3 shows a small insignificant improvement over the CEQE-MaxPool result. CEQE-

MaxPool (fine-tuned) shows the result of using MaxPool with ‘fine-tuned’ contextual embeddings from a BERT model trained for ranking on Robust. The results show small and insignificant differences across all measures. It is almost identical to vanilla embedding effectiveness after being combined with RM3. This indicates that, when used for CEQE-based expansion, pre-trained models are comparable in effectiveness to ones fine-tuned for ranking. Therefore, we did not continue conducting experiments with a fine-tuned model for TREC19-DL and CAR. To our knowledge these are the best unsupervised query expansion results for Robust that do not use external collections.

The SQET supervised method outperforms the baseline BM25, but does not outperform the BM25+RM3 baseline. The SQET-Context_{invRank} outperforms the MASK-QE and SQET, but does not outperform the BM25+RM3 baseline on its own. Examining the results, we hypothesize that this is because of the importance in term weighting with multiple expansion terms. The SQET-Context_{invRank} model is only trained to classify the boundary between relevant expansion terms and non-relevant, and the predicted scores are not effective for term weighting in the query language model. We combine the RM3 and SQET-Context_{invRank} using linear interpolation in SQET-Context_{invRank}Comb, tuned for average precision. This demonstrates that combining the signals from unsupervised RM3 model and supervised SQET result in further gains. The resulting model is significantly better than the RM3 expansion in Recall@100 and Recall@1000 metrics.

Finally, the last row of the table, SQET-Context_{invRank}CEQE-MaxComb, shows the result of the linear interpolation of the CEQE-MaxPool and the SQET-Context_{invRank} tuned on mean average precision and that both of models provide gain in the final results. This model outperforms the BM25+RM3 across nDCG@20, Recall@100, Recall@1000 metrics.

Table 4.3: Ranking effectiveness on the Robust collection. The superscript † and ‡ denote statistical significance over BM25 + RM3 and Static-Embed-PRF, respectively. **Bold** indicates the best value in each section of the table.

Model	P@20	nDCG@20	mAP@1000	Recall@100	Recall@1000
BM25	0.3657	0.4193	0.2574	0.4165	0.6933
BM25 + RM3	0.3998	0.4517	0.3069	0.4610 [‡]	0.7588 [‡]
Static-Embed	0.3675	0.4285	0.2615	0.4217	0.7125
Static-Embed-PRF	0.3781	0.4400	0.2703	0.4324	0.7231
CEQE-Centroid	0.3922	0.4462	0.3019 [‡]	0.4593 [‡]	0.7653 ^{†‡}
CEQE-MulPool	0.3847	0.4360	0.2845 [‡]	0.4517 [‡]	0.7435 [‡]
CEQE-MaxPool	0.4040 [‡]	0.4587	0.3086 [‡]	0.4651 [‡]	0.7689 ^{†‡}
CEQE-MaxPool-RM3Comb	0.4042	0.4577	0.3104 [‡]	0.4656 [‡]	0.7636 [‡]
CEQE-MaxPool(fine-tuned)	0.3986 [‡]	0.4528	0.3071 [‡]	0.4647 [‡]	0.7626 [‡]
MASK-QE	0.3655	0.4223	0.2539	0.4144	0.6940
SQET	0.3695	0.4307	0.2606	0.4231	0.6991
SQET-Context _{invRank}	0.3777	0.4392	0.2835	0.4448	0.7461
SQET-Context _{invRank} -RM3Comb	0.4018	0.4575	0.3127	0.4710 [†]	0.7733 [†]
SQET-Context _{invRank} CEQE-MaxComb	0.4040	0.4611 [†]	0.3140	0.4756 [†]	0.7783 [†]

Table 4.4: Ranking effectiveness of neural ranking on top of query expansion methods for Robust. The superscript † and ‡ indicate significance over BM25 + CEDR and (BM25 + RM3) + CEDR with re-ranking the top 1000, respectively. **Bold** indicates the best value in each section of the table.

Model	P@20	nDCG@20	mAP@1000	Recall@100	Recall@1000
BM25 + RM3	0.3998	0.4517	0.3069	0.4610	0.7588
BM25 + CEDR (MacAvaney et al., 2019)	0.4713	0.5458	0.3312	0.4983	0.6933
(BM25 + RM3) + CEDR	0.4719	0.5435	0.3500 [†]	0.5192 [†]	0.7570 [†]
(BM25 + CEQE-MaxPool) + CEDR	0.4735	0.5462	0.3532 [†]	0.5258 ^{†‡}	0.7719 ^{†‡}
(BM25 + SQET-Context _{invRank}) + CEDR	0.4783	0.5487	0.3475	0.5194	0.7449
(BM25 + SQET-Context _{invRank} RM3Comb) + CEDR	0.4741	0.5437	0.3543 [†]	0.5261 ^{†‡}	0.7722 ^{†‡}

4.4.2 PRF effect on Neural Reranking

We now study how PRF methods impact the effectiveness of neural reranking models. It is important to have effective expansion in the first pass to retrieve sufficient numbers of documents to rerank. The results of our experiments on Robust for unsupervised CEQE as well as supervised SQET-Context models are shown in Table 4.4.

Reranking the results obtained from the expanded queries using the neural reranker such as CEDR (MacAvaney et al., 2019) results in significant gains to average pre-

recision, Recall@100, and Recall@1000 for both RM3, CEQE and SQET-Context. Replacing RM3 with CEQE for expansion results in significant improvement over Recall@100 and Recall@1000. Also, re-ranking the SQET-Context_{invRank} model with CEDR results in highest P@20 and nDCG@20.

4.4.3 Expansion after Reranking

In this section we study how a reranked neural result can be used as a basis for further expansion and reranking (RQ3). This is a critical step because there must be a sufficient number of relevant documents in the top ranks for PRF to be effective. We evaluate multi-round supervised reranking based on expansion runs for Robust for CEQE-MaxPool model in Table 4.5. The top of the table shows results from the leading neural ranking and PRF approaches, including Neural PRF (Li et al., 2018), CEDR, and Birch (Yilmaz, Wang, Yang, Zhang, & Lin, 2019). The results in this section all perform re-ranking on 1000 results from the baseline. We experimented with reranking 100 results and found it consistently performed worse. The baseline model run is BM25+CEDR followed by RM3 expansion with CEDR reranking, which we denote as $(BM25 + CEDR) + RM3 + CEDR$. The results show it outperforms Birch in nDCG@20 and P@20, as well as its own previous result for P@20 on just BM25. Replacing RM3 with CEQE for the expansion consistently outperforms the previous best CEDR results across all measures and significantly over Recall@1000. The runs compare performing RM3 and CEQE-MaxPool on the CEDR baseline (which reranks an initial BM25 first run). The second pass results are then reranked again using CEDR. The result has further improvement over previous approaches. The same trend continues, with the CEQE-MaxPool outperforming the reranked RM3 run.

A common approach when applying BERT-based neural ranking is to perform learning-to-rank to combine the BERT and retrieval score. A simple proven approach

Table 4.5: Ranking effectiveness of multi-round neural re-ranking and expansion for Robust. The superscript † and ‡ indicate significance over BM25 + CEDR and (BM25 + CEDR) + RM3 baselines, respectively.

Model	P@20	nDCG@20	mAP@1000	Recall@100	Recall@1000
Neural PRF-DRMM (Li et al., 2018)	0.4064	0.4576	0.2904	-	-
BM25 + CEDR (MacAvaney et al., 2019)	0.4713	0.5458	0.3312	0.4983	0.6933
Birch (Yilmaz, Wang, et al., 2019)	0.4657	0.5325	0.3697	-	-
(BM25 + CEDR) + RM3	0.4458	0.5211	0.3321	0.4881	0.7751 [†]
(BM25 + CEDR) + RM3 + CEDR	0.4783	0.5499	0.3574 [†]	0.5291 [†]	0.7751 [†]
(BM25 + CEDR) + RM3 + CEDR Interp	0.4837 [†]	0.5565	0.3739 [†]	0.5440 [†]	0.7751 [†]
(BM25 + CEDR) + CEQE-MaxPool	0.4504	0.5250	0.3366	0.4931	0.7874 ^{†‡}
(BM25 + CEDR) + CEQE-MaxPool + CEDR	0.4799	0.5516	0.3601 [†]	0.5332 [†]	0.7874 ^{†‡}
(BM25 + CEDR) + CEQE-MaxPool + CEDR Interp	0.4904[†]	0.5621[†]	0.3773[†]	0.5486[†]	0.7874^{†‡}

is linear interpolation of the underlying retrieval score with neural ranking model (W. Yang et al., 2019; Yilmaz, Wang, et al., 2019). We apply this to the two best runs, learning the interpolation using the previously described cross-validation setup. The results demonstrate that linear interpolation with these expansion runs continues to show gains. The interpolation with CEQE-MaxPool is the best performing, and compared with the previous Birch shows over 5% relative gain P@20 and nDCG@20 as well as improving MAP. These results show that multiple rounds of expansion and reranking can continue to result in significant improvements.

4.4.4 Intrinsic Expansion Evaluation

We examine the effectiveness of the expansion approaches to rank positive expansion terms that improve Mean Average Precision (at 1000) when added to the query. This experiment evaluates a method’s ability to identify good expansion terms in isolation. The results are shown in Table 4.6 for the key expansion models to compare for Robust. Since a fixed top-k expansion terms are usually selected for expansion we evaluate the intrinsic evaluation with set-based precision numbers at common thresholds for the number of expansion terms. The results show that a well-tuned Relevance Model significantly outperforms query expansion models based on static embeddings. In contrast, we find that CEQE provides improvements in early ranks for

P@10 and P@20. All the CEQE models significantly improve over static embedding models across all metrics. And further, we find that CEQE-MaxPool significantly outperforms the Relevance Model expansion effectiveness for P@10 and P@20. It is insignificantly different from the Relevance Model at rank 100. This indicates that the strength of CEQE is selecting a higher number of “good” terms earlier, allowing improved effectiveness with fewer expansion terms.

The SQET-Context_{invRank} is the best performing model in the early ranks among all models, but is slightly outperformed by SQET-Context_{wSum} at rank 100. The SQET model is significantly outperformed by the Relevance Model, SQET-Context_{wSum} and SQET-Context_{invRank} across all measures. This indicates the power of BERT when it is provided with the context and term relations. Moreover, SQET-Context_{Max} is also outperformed by Relevance Model, SQET-Context_{wSum} and SQET-Context_{invRank}. This shows that the different context around the candidate term across the corpus provides valuable information for the ranking of the term. Also, SQET-Context_{invRank} outperforms the BM25_{invRank} highlighting the effect of BERT scores in the ranking. The poor performance of the MASK-QE demonstrates that since the pre-trained model is ranking all the terms in its vocabulary, it is a noisy model and cannot generate a good ranking of expansion terms for the target corpus. We investigate the effect of combining the knowledge coming from our unsupervised model, CEQE-MaxPool and our supervised model, SQET-Context_{invRank} by calculating the Reciprocal Rank Fusion (RRF) of their expansion terms ranking. The RRF(CEQE-MaxPool, SQET-Context_{invRank}) significantly outperforms the Relevance Model across all measure. This shows that both of the two methods provide valuable and different signals in ranking expansion terms. Also, the RRF improves upon both CEQE-MaxPool and SQET-Context_{invRank} across the P@20 and P@100.

Table 4.6: Intrinsic ranking evaluation of positive expansion terms on Robust. The superscript † denotes the statistical significance over the Relevance Model. **Bold** indicates the best result in each column.

Model	P@10	P@20	P@100
Relevance Model	0.1693	0.1419	0.0871
Static-Embed	0.1008	0.0780	0.0511
Static-Embed-PRF	0.1357	0.1083	0.0655
CEQE-MulPool	0.1349	0.1174	0.0737
CEQE-Centroid	0.1751	0.1481	0.0826
CEQE-MaxPool	0.1830 [†]	0.1500 [†]	0.0841
MASK-QE	0.0544	0.0515	0.0414
SQET	0.1207	0.1104	0.0758
SQET-Context _{Max}	0.1332	0.1085	0.0695
SQET-Context _{wSum}	0.1763	0.1556 [†]	0.0921
SQET-Context _{invRank}	0.1942[†]	0.1560 [†]	0.0900
BM25 _{invRank}	0.1610	0.1336	0.0802
RRF(CEQE-MaxPool, SQET-Context _{invRank})	0.1938 [†]	0.1583[†]	0.0976[†]

4.5 Qualitative Behavior Analysis

Query-by-Query Analysis. To better understand the ranking behaviour of our proposed model, we compare the top ranked expansion terms of RM1, CEQE-MaxPool and SQET-Context_{invRank} in Table 4.7. We illustrate the performance of our approach using [Topic 405, cosmic event] and [Topic 685, oscar winner selection] which performed well in the extrinsic evaluation (more than 10% improvement of mAP when comparing CEQE-Max and BM25+RM3). The first row has the terms (unstemmed) with the greatest improvement for the query.

We observe that the CEQE model identifies mostly all of the positive terms from RM as well as introducing additional relevant terms for both topic 405 and 685. More generally, we see that the CEQE terms appear to have a stronger semantic relationship with the query terms. The RM terms appear most loosely related and have additional noise terms, including general terms like ‘article’, ‘large’ and ‘type’ for topic 405. We hypothesis, this is because RM focuses on terms that co-occur across multiple

Table 4.7: Example query expansion terms for Topic [405 , cosmic events] and [685, oscar winner selection] in Robust collection. This includes the important intrinsic positive labels, Relevance Model, CEQE-MaxPool and SQET-Context_{invRank} expansion terms. Terms with positive intrinsic labels are bolded.

Topic 405	cosmic events
Positive terms:	astronomers, astronomical, bang, big, galaxies, light, matter, particle, particles, physicist, scientists, space, theory, universe, years
RM:	energy, space , solar, particle , earth , radiation, proton, article, ray, large, universe , type, fluence, magnitude, particles
CEQE-MaxPool:	space , universe , radiation, energy, earth, solar, particles , big , years , matter , dust, article, ray , bang , galactic , scientists
SQET-Context:	universe , years , astronomers , radiation, scientists , bang , matter , galaxies , dust, energy, physicist , time, big , research, theory , astronomical
Topic 685	oscar winner selection
Positive terms:	academy, academys, nominations, nomination, critics, members, award, awards, branch, ignored, true, films, film, directors, director, filmmaker
RM:	best, film , picture, million, academy , years, award , home, edition, films , man, four, 1, 5
CEQE-Maxpool:	film , academy , picture, winners, award , films , million, oscars, box, presented, awards , director , years, nominations
SQET-Context:	awards , oscars, nominations , nominees, years, edition, award , nominated winners, home, films , dga, winning, film , academy , ua, nomination

PRF documents, but it does not explicitly model the relationship to the query. In contrast CEQE explicitly focuses on the query. As a result, the CEQE model produces fewer terms that co-occur by chance. Further, for topic 405 SQET-Context_{invRank} ranks more positive expansion terms in higher ranks in comparison with RM1. Also, the SQET-Context_{invRank} rank two terms ‘astronomers’ and ‘astronomical’ in the top ranks that both RM and CEQE-MaxPool have missed. Moreover for topic 685, SQET-Context_{invRank} is able to exclude the digits, while RM is ranking them in top expansion terms.

Further, Table 4.8 shows the win/loss comparison to BM25 for three expansion methods: BM25+RM3, CEQE-MaxPool and SQET-Context_{invRank}. The CEQE-MaxPool has the highest wins across three methods. However, CEQE-MaxPool and BM25+RM3 have similar behavior with losses. The SQET-Context_{invRank} model

Table 4.8: Win/Loss comparison to BM25 on Robust.

Model	Win	Neutral	Loss
BM25	-	-	-
BM25 + RM3	151	26	73
CEQE-MaxPool	154	23	73
SQET-Context _{invRank}	149	32	69

alleviates the losses by using supervision, but it is more conservative and has the highest neutrals.

4.5.1 Computational Cost Analysis

We use a BERT-based model to produce contextualized embeddings for query expansion, which incurs similar computational costs as BERT-based reranking methods like CEDR (MacAvaney et al., 2019). Generating these embeddings is the most computationally-intensive step of all such methods. Compared to other query expansion approaches, our work’s computational costs are most similar to BERT-QE’s (Zheng et al., 2020). Both approaches consist of a query expansion step that requires processing with a BERT model followed by an (optional) reranking step that again processes the top document with a BERT model. The core focus of this work is on effectiveness, although efficiency is an important area for future research.

4.6 Discussion

In this section, we discuss subsequent research to our work that utilizes Transformer models for leveraging the context and semantic information within Pseudo Relevant Feedback (PRF) documents and enhance query representation for better information retrieval. Additionally, we explore the pioneering studies that utilize large language models (LLM) like GPT3.5 for query expansion.

Concurrently and following our research on query expansion for sparse retrieval, researchers have investigated incorporating pseudo-relevance feedback (PRF) techniques to enhance query representations in dense retrieval. Notably, H. Yu et al. (2021) proposed ANCE-PRF, which combines the original query with PRF passages retrieved from an ANCE (L. Xiong et al., 2020) model and encodes them using a BERT architecture to train a new query encoder for improved dense retrieval ranking. Additionally, X. Wang et al. (2021) introduced ColBERT-PRF, a vector-based PRF approach that refines ColBERT’s query-document scoring function by clustering and selecting discriminative embeddings from pseudo-relevant documents, ultimately integrating them with ColBERT’s original scoring mechanism to derive a final relevance score.

More recently, Mackie et al. (2023b) proposed a LLM-based approach by leveraging GPT-3 to generate diverse query-specific text formats, such as keywords, entities, chain-of-thought reasoning, facts, news articles, and essays. They demonstrated that combining these generated text types outperforms traditional sparse retrieval methods like BM25 across multiple datasets. Furthermore, their subsequent work (Mackie et al., 2023a) showed that integrating this generative expansion technique with established pseudo-relevance feedback (PRF) methods leads to even greater improvements, highlighting the complementary strengths of these two approaches.

4.7 Summary

In this chapter, we explored how Transformer models can be employed for the task of query expansion for more effective information retrieval systems. We built upon the Relevance Model (RM) approach, using the top-k retrieved documents in the first pass retrieval as pseudo-relevant documents and focused on leveraging the *context* within the pseudo-relevance feedback models to shift away from word-count approaches. Transformer models with their inherent attention structure allowed us

to incorporate the context within these documents, leading to enhanced ranking of relevant expansion terms to user’s original query.

We developed methodologies in both unsupervised and supervised frameworks. Our unsupervised approach, CEQE, utilizes BERT embeddings to calculate the similarity between a term in a pseudo-relevant document and the query, considering the term’s context. We show that CEQE outperforms static embedding methods in terms of MAP and Recall and performs at least as well as the strong word-based feedback model, RM3 on multiple collections (*Contribution 2.1*). We further show that neural reranking combined with CEQE results outperforms previous approaches in terms of MAP and Recall (*Contribution 2.2*).

Our supervised model, SQET, formulates query expansion as a classification task and leverage Transformer models in a cross-attention architecture to label an expansion term relevant to the query or non-relevant. The predicted relevancy score is then used to rank the expansion term and representing how relevant each term is to the original query. The performance of SQET variants is on par with term frequency-based feedback models. Moreover, a hybrid approach combining SQET with RM3, a term frequency-based model, shows additional improvements over using either method separately (*Contribution 2.3*).

CHAPTER 5

EXAMPLE DOCUMENTS AS THE CONTEXTUAL INFORMATION SOURCE

Statement of Contribution

Part of the work presented in this chapter appeared as a full paper in ICTIR 2023 as Huang, Naseri, Bonab, Sarwar and Allan where Huang and I were joint first authors. The large-scale training dataset, Wiki-QBE, the KeyPhrase model results on the Wiki-QBE dataset and SummPip model results on all the QBE datasets are contributed by Huang.

The task of Query-By-Examples (QBE), where the users express their information need by providing single or multiple examples instead of formulating an exact query is a widespread scenario in professional and domain-specific search such as legal case retrieval (Abolghasemi et al., 2022; Althammer et al., 2022; Askari & Verberne, 2021; M.-Y. Kim et al., 2019; Shao et al., 2020), scientific literature retrieval (Cohan et al., 2020; Mysore et al., 2021), and patent retrieval (Fujii et al., 2007; Piroi & Hanbury, 2019). Most of the prior works target the scenario where there is only one example document, whereas having multiple example documents is a more complicated task in terms of input representation.

The core research question of this chapter is, “How can the rich context embedded in query example documents be leveraged to retrieve a ranked list of documents that are relevant to users’ information needs?”. To address this research question we leverage capabilities of Transformer architecture in capturing context within a text.

This chapter concentrates on developing multiple end-to-end architectures for the task of query by example retrieval, integrating Transformer-based models to specifically focus on the query and document representation and improving the performance of information retrieval systems.

In Section 5.1 we construct the dataset for the QBE task. In Section 5.2, we present our retrieval strategy as a neural re-ranking strategy. In Section 5.3 we introduce our Passage-based Relevancy Representation with Multiple Examples (PRRIME) model. In Section 5.4 we present the cross-encoder neural re-ranking strategy for the task of QBE. In Section 5.5 we introduce a multi-task learning strategy for re-ranking a list of retrieval results as well as generating the exact information need.

5.1 Query by Example Retrieval Datasets

We build multiple Query By Example (QBE) datasets based on the fact that both query example documents and the labeled relevant documents are relevant to the

users’ information need. Therefore, we leverage existing keyword-based datasets and build the input example documents by sampling from the documents judged relevant to the query. Given a keyword query, we randomly sample N (defined as the query length) relevant documents as the query and leave the rest of the relevant documents as the retrieval targets. Following this approach we built two evaluation dataset: 1) Robust04-QBE and 2) Multi-News-QBE. We select query topics with more than five relevant documents and randomly sample 1 to 5 relevant document and build 5 datasets with 1 to 5 query-documents, respectively. The sampled example documents in the dataset with k query-documents is a subset of example documents with *more than k* query-documents. As an example, from the dataset with 1 query-document to the dataset with 2 query-documents only one new document is introduced for each query topic. Table 5.1 shows the statistics of the evaluation datasets, along with the along with the large-scale training dataset, Wiki-QBE.

Robust04-QBE. It is based on the standard ad-hoc retrieval dataset, Robust04 a high quality and well-studied IR collection for both traditional retrieval models and neural retrieval models (J. Lin, 2019), where the corpus consists of 528K newswire articles. Robust04 has 250 query topics, among them 233 have more than five relevant documents.

Multi-News-QBE. It is based on the Multi-News (Fabbri, Li, She, Li, & Radev, 2019) dataset, a large-scale multi-document summarization dataset. The Multi-News dataset originally is constructed from the articles of the site newser.com which are human-written summaries of multiple news articles. The human-written summaries are used as the target text sequence and the news articles cited (i.e., linked) in the summary articles are used as source text sequence in the summarization task. To build our QBE retrieval dataset, first we collect all the source news articles linked in the newser.com summary article to create our corpus. We define the source news articles of a summary article relevant to each other and each one of them can be a

Table 5.1: Statistics of proposed QBE datasets. Avg $\#d^+/q$ denotes the average number of relevant documents per query.

	Wiki-QBE	Robust04-QBE	Multi-News-QBE
Document count	2.4M	528,155	135,980
Query count	183,837	233	1,036
Query documents	3	{1, 2, 3, 4, 5}	{1, 2, 3, 4, 5}
Avg. $\#d^+/q$	35.10	70.08	6.69

query-document that is covering a query topic which is the summary article. However, to avoid misguiding our retrieval approaches and decrease noise we delete query topics where at least one of its source document appeared as a source document in other query topics. Further, we perform text processing on the news article and remove lines that are social media related text such as “*Follow us on Facebook*”, texts and error messages regarding the crawling and fetching webpages like “*JavaScript is disabled ...*” and finally placeholder texts such as “*Looking for news you can trust?*” and “*Subscribe to our free newsletters.*”

WikIR-QBE. It is a weakly supervised QBE dataset and is built following the idea of the WikIR (Frej, Schwab, & Chevallet, 2020). In particular, the content of each Wikipedia article can be used as an example for its title since each article is relevant to its title. Huang, Naseri, Bonab, Sarwar, and Allan (2023) assumes that if an article a contains an internal link to another article a_t in its first sentence and the anchor text exactly matches the title of article a_t , then the content of article a can also be used as an example for a_t ’s title. Finally, for titles with more than five content examples, three examples are randomly sampled as query documents and the other ones are used as relevant documents. Lastly, the title and the first sentence of each article are removed, since all the information used to build the examples for the title is contained in the first sentence of the articles.

5.2 Neural Re-ranking in Query by Example Retrieval

We employ the neural re-ranking strategy in which we first obtain an initial set of candidate documents using a keyword matching retrieval technique, query likelihood, and then re-rank the small set of candidate documents using the neural models discussed in Sections 5.3, 5.4, and 5.5.

5.2.1 First Stage Retrieval

To obtain the initial query for the first stage of retrieval we adopt three different approaches: 1) Keyphrase, 2)SummPip, and 3)docT5query.

Keyphrase. We concatenate query documents into one long sequence and extract keyphrases as the query. We select 1- to 3-grams as the candidate phrases, ranked based on their TF-IDF scores¹, and select the top 100 keyphrases with their corresponding weight. We use Galago’s² query language model and its implementation of query likelihood model with the default parameter to retrieve the documents from the collection.

SummPip. We leverage an off-the-shelf unsupervised multi-document summarization model, SummPip (Zhao et al., 2020), to generate a summary for the query example documents that serves as the query for the term-matching information retrieval technique. Galago’s query likelihood model with default parameters is then used to retrieve ranked list of documents.

docT5query. To explore if the current state-of-the-art question generation methods can be used to identify the hidden information need (i.e., query) given the query-documents we take advantage of the doc2query model (Nogueira et al., 2019), a T5 based sequence-to-sequence model trained on (question, passage) pairs from the MS-Marco passage collection (Nguyen et al., 2016), to generate candidate questions. Since

¹We use <https://github.com/boudinfl/pke> implementation for TF-IDF scoring.

²<https://www.lemurproject.org/galago.php/>

Table 5.2: Query extraction methods for the first stage retrieval. For each column, the highest value is marked with bold text. At this stage, we select R@100 as the primary evaluation metric. Subscripts refer to the standard deviation of 5 corpuses.

Method	Wiki-QBE			Robust04-QBE			Multi-News-QBE		
	MAP	MRR	R@100	MAP	MRR	R@100	MAP	MRR	R@100
Keyphrase	0.1803 _(0.0056)	0.3970 _(0.0095)	0.4812 _(0.0090)	0.1358 _(0.0060)	0.5377 _(0.0202)	0.3127 _(0.0101)	0.4080 _(0.0032)	0.6089 _(0.0085)	0.7738 _(0.0035)
SummPip	0.1367 _(0.0066)	0.3298 _(0.0094)	0.3688 _(0.0116)	0.1002 _(0.0064)	0.4464 _(0.0210)	0.2405 _(0.0088)	0.3917 _(0.0060)	0.5987 _(0.0120)	0.7538 _(0.0073)
docT5query	0.1502 _(0.0043)	0.3516 _(0.0049)	0.4110 _(0.0068)	0.1353 _(0.0056)	0.5191 _(0.0175)	0.3088 _(0.0076)	0.3674 _(0.0030)	0.5590 _(0.0093)	0.7424 _(0.0028)

the input of the doc2query model is passage length we break down our documents into passages and using the doc2query model we generate questions for each passage. Next, we select the top 10 questions generated for each passage and concatenate them to each other and build a mid-point representation for the query-documents. Then, similar to the Keyphrase approach we extract the keyphrases from the mid-point representation and rank them based on their TF-IDF scores. Finally, we select the top 100 keyphrases and their corresponding weights and perform the retrieval using Galago’s query likelihood implementation with its default parameters.

Table 5.2 shows the results of our first-stage retrieval methods. We can see that the Keyphrase approach outperforms the SummPip and docT5query methods across all measure for all datasets. Therefore, we select Keyphrase as our approach for first-stage retrieval, obtaining the initial set of candidate documents.

5.3 Passage-based Relevancy Representation with Multiple Examples (PRRIME)

Employing pretrained language models based on the Transformer architecture (Vaswani et al., 2017) in neural retrieval models resulted in the state-of-the-art performance in the text ranking task (J. Lin, Nogueira, & Yates, 2021). Further, scaling up Large Language Models (LLMs) by pretraining larger decoder-only models on larger and higher quality corpora resulted in impressive effectiveness in few-shot or zero-shot text generation. As a result researchers have modeled ranking as a text generation

task to generate a reordered list of candidates (Ma, Zhang, et al., 2023; Pradeep et al., 2023; Sun et al., 2023) or alternatively comparing documents (Qin et al., 2023) in a pairwise setting. However, pretrained language models often face limitations due to their fixed context-window sizes, which makes processing long documents challenging. For instance, BERT, the first widely-adopted pretrained language model, is constrained to 512 tokens. Even more recent models, such as those in the GPT family (including GPT-3.5 and GPT-4), with their expanded context windows ranging from 4096 to 128,000 tokens, experience performance loss in downstream tasks like question answering across multiple documents, when relevant information is located in the middle of a long context (N. F. Liu et al., 2023). This limitation makes leveraging pretrained language model for the task of query by example retrieval more challenging as this task requires representing one or multiple documents for the query along with the candidate document for ranking.

Previous work in ad-hoc retrieval given a keyword query, alleviated the fixed context-window by breaking down the candidate documents into sentences (Yilmaz, Yang, Zhang, & Lin, 2019) or passages (Y. Kim et al., 2021), scoring them independently with respect to the query, and aggregating the scores to compute the final scores of the documents. Li et al. (2020) propose a model that aggregates passage relevancy *representation* instead of aggregating the passage-based relevancy *score*.

Inspired by this line of work, we investigate how to combine passage-level relevancy to get the final document-level score in the QBE problem. The QBE task is more challenging because the query consists of multiple documents instead of a single keyword, and to effectively utilize Transformer models, we need to break down both example and candidate documents into passage-length inputs. Our initial experiments show that merely aggregating the similarity scores between query-passage and document-passage pairs results in sub-optimal performance.

Given the poor performance of relevance score aggregation, we investigate aggregating passage-level relevancy representations to achieve a final relevancy score between query-example documents and the candidate document. Figure 5.1 shows the architecture of our model, **P**assage-based **R**elevancy **R**epresentation w**I**th **M**ultiple **E**xamples (PRRIME).

In general, given a set of example documents as the query Q and a candidate document D , due to limited context-window size we break down each document to passages that can be handled by a Transformer architecture individually. To do so, a sliding window of k words is applied to the document with a stride of w words. More formally, $Q = \{P_{1Q_1}, P_{2Q_1}, \dots, P_{|Q_1|Q_1}, P_{1Q_2}, \dots, P_{|Q_m|Q_m}\}$ where P_{iQ_j} is the i -th passage of query document j and $|Q_j|$ is the number of passages in the query document j . And $D = \{P_1, P_2, \dots, P_{|D|}\}$ where $|D|$ is the number of passages in the document D and P_j is the j -th passage of document D .

Afterwards, we form pairs by concatenating each passage from the set of query document passages Q with each passage from the set of candidate document passages D . Each of these concatenated pairs is then treated as an individual input for the model.

In particular with the BERT Transformer architecture the input is as follows:

Input: [CLS] P_{iQ_j} [SEP] P_k

where [SEP] and [CLS] are special tokens which help the model distinguish between different input segments and identify the overall representation of the input sequences.

To be more specific, the corresponding output of the [CLS] token in the last layer is parameterized as a relevance representation, $r_{P_k}^{P_{iQ_j}}$, between P_{iQ_j} and P_k .

Given the passage relevance representations, $R = \{r_{P_1}^{P_{1Q_1}}, r_{P_2}^{P_{1Q_1}}, \dots, r_{P_k}^{P_{iQ_j}}, \dots\}$, PRRIME summarizes R into a single dense representation using a robust max pooling operation. Max pooling, widely used in Convolutional Neural Networks (CNNs) (Scherer, Müller, & Behnke, 2010), effectively extracts position-invariant features.

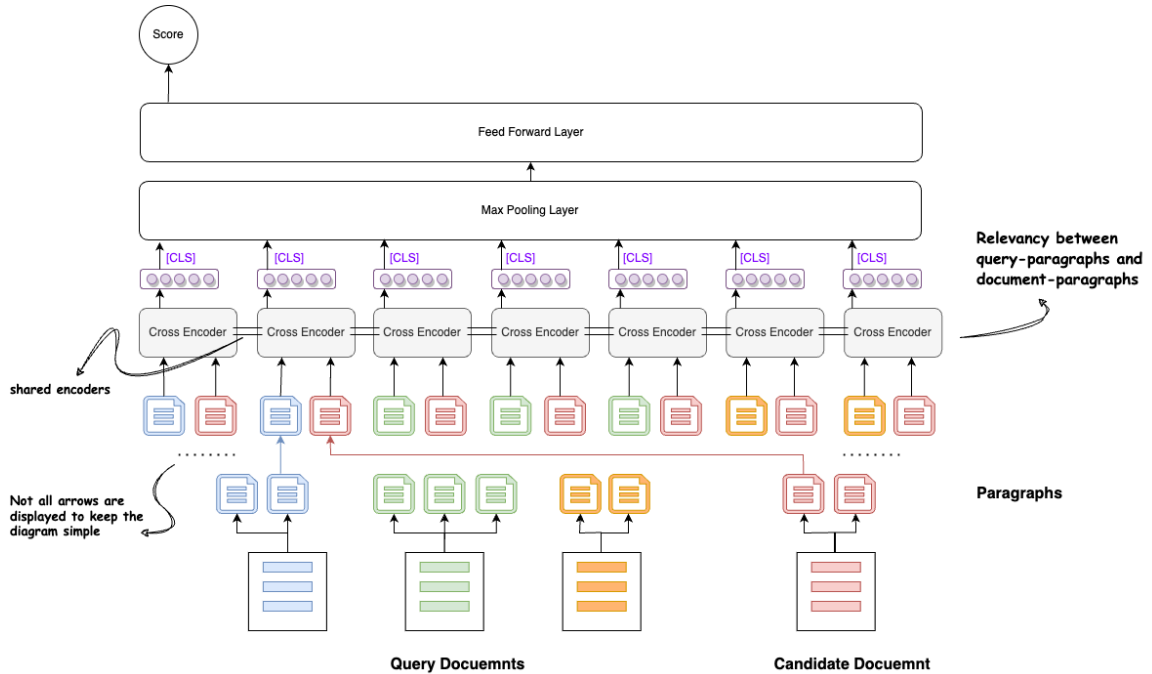


Figure 5.1: Overview of the PRRIME model.

The final relevancy score is calculated by linearly transforming the relevance representation into a scalar using a feed-forward layer.

System Details. We focus on using 3 example documents as the query in our experiments. We take advantage of a pre-trained Transformer-based semantic search model, in particular a BERT model fine-tuned on the MSMarco dataset (Bajaj et al., 2016)¹. For all our datasets, we further fine-tune this checkpoint model for query-by-example re-ranking. We parse the input sequence into paragraphs with a fixed length of 250 words and a stride between paragraph of 50 words. We use the relevant documents as the positive samples and randomly sample negative documents from the ranked list of the first stage retrieval to form training triplets. We train for 40 epoch with a batch size of 64.

¹<https://huggingface.co/Capreolus/bert-base-msmarco>

Given the constraints of limited GPU memory, we adopt a strategy that involves working with a fixed number of passages from both query and candidate documents.

- **PRRIME-Adhoc:** This approach selects the first, last, and a set of randomly chosen passages from the middle of the document (for each example document and candidate document) to provide a broad representation.
- **PRIMME-Summ:** This approach leverages Large Language Models (LLMs), in particular GPT3.5-Turbo, in a zero-shot setting to generate a shorter text representation. This shorter text representation addresses the common information need among query examples, the different aspects discussed in each example, and the questions answered by each. Due to the 16,385 token limit of GPT3.5-Turbo's context window, we restrict each example document to 5,000 tokens and truncate any exceeding that length.

We use the following system and user prompts for the GPT3.5-Turbo chat completion model.

System Prompt:

You are provided with three example documents. Analyze the provided documents to identify the overarching information need, as well as the specific contributions and questions addressed by each document. Your response should have the following sections "summary", "aspects" and "questions". The definition of each section is as follow:

-- "summary": Briefly summarize each document in no more than 5 sentences. Clearly state the shared information need that connects all documents.

-- "aspects": Determine how does each individual document contribute to addressing this common information need. Be specific about the unique aspects or angles covered.

-- "questions": Address what specific questions within the broader information need does each document answer. Write 3 questions for each document.

User Prompt:

The three example documents are as follow:

Document1: [Document 1]

Document2: [Document 2]

Document3: [Document 3]

Evaluation Metrics. For evaluating retrieval effectiveness at the reranking stage, we report mean average precision (MAP), mean reciprocal rank (MRR), and precision of the top 10 retrieved documents (P@10).

Results. Table 5.3 shows the result of PRRIME variants on the three datasets of QBE that we presented in section 5.1. PRRIME-Adhoc statistically significantly outperform the Keyphrase baseline and PRRIME-Summ on both Wiki-QBE and Multi-News-QBE across all evaluation metrics. While PRRIME-Adhoc demonstrates competitive performance on Robust04-QBE for MAP and P@10, it outperforms the baseline in Mean Reciprocal Rank (MRR). PRRIME-Summ also statistically outperforms Keyphrase on all metrics for Wiki-QBE and on MAP and P@10 for Multi-News-QBE. We hypothesize that the small number of queries (approximately 200) used during fine-tuning limits the PRRIME variants' ability to outperform the Keyphrase baseline. All in all the improvement achieved by the PRRIME variants shows the effectiveness of aggregating relevancy representation for the task of query by example retrieval.

Table 5.3: PRRIME Model performance on QBE datasets. Subscripts refer to the standard deviation of 5 corpuses. For PRRIME-Adhoc, statistically significant improvements are marked by \star (over Keyphrase), \blacktriangle (over PRRIME-Summ).

Model	Wiki-QBE			Robust04-QBE			Multi-News-QBE		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Keyphrase	0.1803 _(0.0050)	0.3970 _(0.0085)	0.1635 _(0.0038)	0.1371 _(0.0058)	0.5376 _(0.0202)	0.2957 _(0.0088)	0.4081 _(0.0032)	0.6090 _(0.0085)	0.1871 _(0.0005)
PRRIME-Summ	0.2114 _(0.0033)	0.4552 _(0.0157)	0.19033 _(0.0036)	0.1355 _(0.0041)	0.5260 _(0.0158)	0.3035 _(0.0071)	0.4209 _(0.0019)	0.6076 _(0.0029)	0.1985 _(0.0019)
PRRIME-Adhoc	0.2457 \star _(0.0023)	0.5080 \blacktriangle _(0.0110)	0.2193 \blacktriangle _(0.0026)	0.1297 _(0.0053)	0.5426 _(0.0122)	0.2881 _(0.0089)	0.4883 \blacktriangle _(0.0085)	0.6937 \blacktriangle _(0.0107)	0.2243 \blacktriangle _(0.0025)

5.4 Cross-Encoder Reranking in Query Example Retrieval

We study cross-encoder neural reranking in query by example retrieval where we concatenate query-documents and the candidate document as input to the Transformer network. However, since the input length of concatenation of multiple documents is generally longer than input sequence length of conventional Transformer models such as BERT, we employ Longformer (Beltagy et al., 2020), a long-sequence Transformer model. Longformer utilizes a local windowed attention mechanism with a task motivated global attention as a replacement for the standard self-attention which enables it to process inputs with thousands of tokens. We take advantage of the Longformer model introduced by Caciularu et al. (2021) which is pretrained on a Multi-Document corpus (Fabbri et al., 2019) with the goal of capturing cross-text relationships, particularly aligning or linking matching information elements across documents. We refer to it as **Cross-Document Longformer** (CD-Longformer) in our table of results. Following Caciularu et al. (2021), we tagged sentences of each document with begin (`<s>`) and end of sentence(`</s>`) tokens as well as labeling begin and end of documents with the special tokens of begin (`<doc-s>`) and end of document (`</doc-s>`). Further, we differentiate the query-documents and the candidate document by special tokens of `<query>` and `<doc>`. Figure 5.2 shows the architecture of cross-encoder reranking using special tokens for query by example retrieval. It is worth mentioning that exceeding the Transformers input size is inevitable, therefore for query-documents we set a limit of 2600 tokens and truncate the longest document

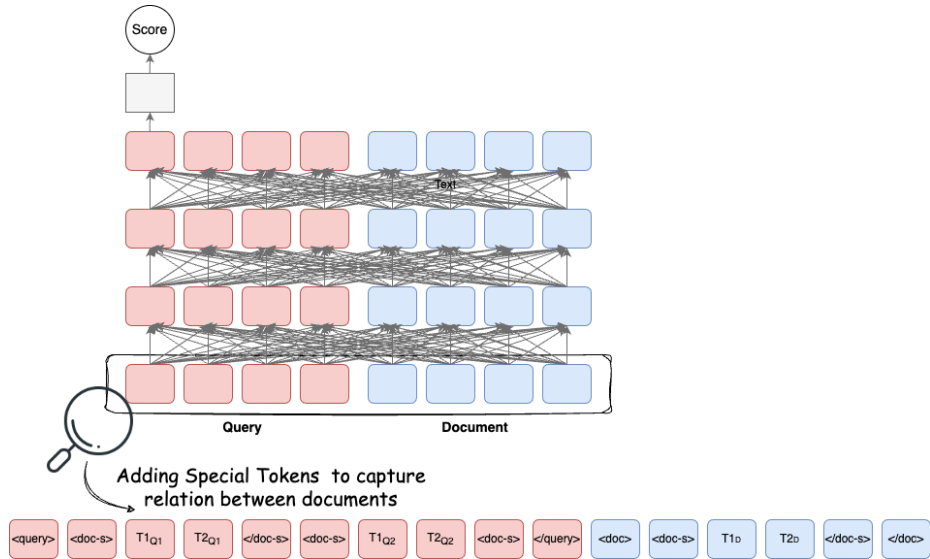


Figure 5.2: Neural reranking using cross-encoder architecture in query by example retrieval

among the query example documents in case of passing this limit. Further, since the candidate document is only one documents in comparison with the query-documents that are multiple documents we set the maximum token length to 1400 tokens.

System Details. We use AdamW optimization algorithm (Loshchilov & Hutter, 2017) with a learning rate of $2e-5$ for training. We use the positive documents and randomly sample negative documents from the ranked list of the first stage retrieval to form training tuples. First, we train on the Wiki-QBE dataset with a batch size of 24 for 22 epochs. Then we fine-tune it on Robust04-QBE and MultiNews-QBE.

Evaluation Metrics. For evaluating retrieval effectiveness at the reranking stage, we report mean average precision (MAP), mean reciprocal rank (MRR), and precision of the top 10 retrieved documents (P@10).

Results. Table 5.4 shows that the CD-Longformer outperforms the baseline statistically significantly by approximately 24% in terms of MAP@100 on the Wiki-QBE dataset. However, it performs worse than the baseline on the evaluation datasets Robust04-QBE and Multi-News-QBE. We hypothesize the reason is the CD-Longformer

Table 5.4: Cross-encoder reranking results on QBE datasets. Subscripts refer to the standard deviation of 5 corpuses. Statistically significant improvements of CD-Longformer are marked by \star over Keyphrase.

Model	Wiki-QBE			Robust04-QBE			Multi-News-QBE		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Keyphrase	0.1803 _(0.0050)	0.3970 _(0.0085)	0.1635 _(0.0038)	0.1371 _(0.0058)	0.5376 _(0.0202)	0.2957 _(0.0088)	0.4081 _(0.0032)	0.6090 _(0.0085)	0.1871 _(0.0005)
CD-Longformer	0.2234* _(0.0059)	0.4715* _(0.0148)	0.2019* _(0.0054)	0.0963 _(0.0031)	0.3994 _(0.0125)	0.2231 _(0.0044)	0.3143 _(0.0044)	0.4882 _(0.0031)	0.1629 _(0.0017)

model is tuned on the latent characteristics of Wiki-QBE documents and the small number of samples in the evaluation datasets is not enough enough to compensate when fine-tuning on them.

5.5 Multi-task Query Generation and Re-ranking in Query by Example Retrieval

Auxiliary training paradigm focuses on transferring knowledge from auxiliary tasks to improve the target recommendation task. While multi-task learning aims to improve the performance across all tasks, auxiliary learning differs in that high test accuracy is only required for a primary task, and the role of the other tasks is to assist in generalization of the primary task. Auxiliary learning has been widely used in many areas. For example in ranking, Ju, Yang, and Wang (2021) leverage the auxiliary task of query generation for passage ranking. Further, Abolghasemi et al. (2022) utilized auxiliary task of representation learning to improve the performance of re-ranking in query by example retrieval task.

By building on the auxiliary training paradigm, we develop an end-to-end re-ranking system for query-by-example retrieval, integrating query generation as an auxiliary task. Our objective is to enhance the performance of our primary task, re-ranking candidate documents, by training it alongside with the query generation task within a multi-task network framework. This strategy enables the transfer of critical knowledge from the auxiliary task, notably the user’s exact information needs, to

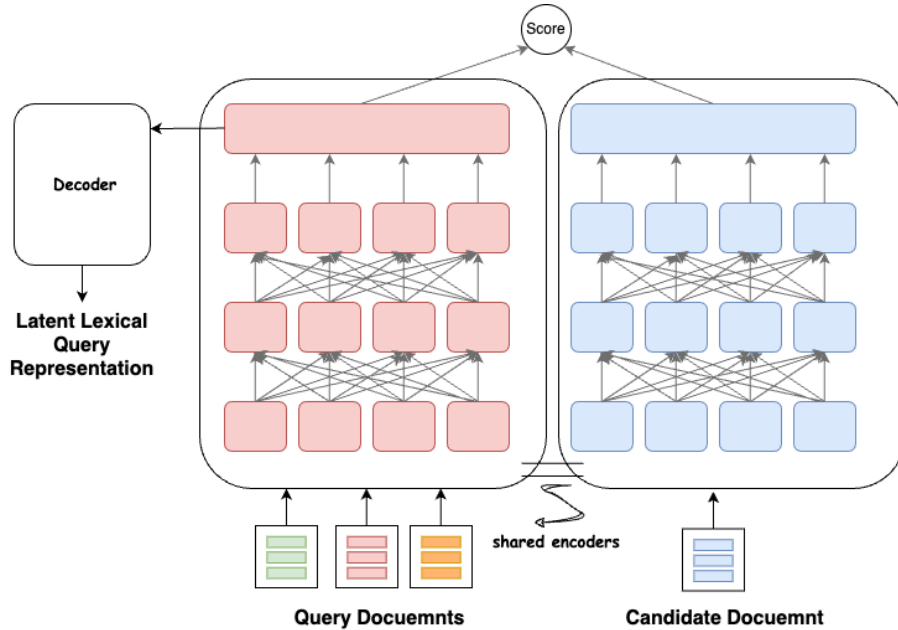


Figure 5.3: Overview of the multi-task learning framework.

the target task, resulting in more robust shared feature representations that improve document re-ranking. Additionally, the inclusion of query generation enhances model explainability, offering clearer insights into the final query representation at test time. Figure 5.3 depicts a high-level architecture of our model. The architecture of query documents encoder and candidate document encoder is based on the Transformer model architecture, in particular Longformer (Beltagy et al., 2020) that has a context window size of 4096 tokens. Additionally, we utilize architecture and the weights from the pre-trained decoder module of the Longformer-Encoder-Decoder (LED) model to structure our decoder architecture, initializing it with LED weights.

Objective Functions. Given a set of example documents as the query Q and a candidate document D , we define the objective functions for each task as follows:

Document Ranking: We adopt the Multiple Negative Ranking Loss (Henderson et al., 2017) to train the query documents and candidate document encoders. This loss expects a query Q and a positive document D^+ and a set of negative documents $\{D_1^-, D_2^-, \dots, D_N^-\}$. For efficiency and simplicity we use the positive documents of

other queries in a training batch of stochastic gradient descent as negative documents for the current query.

$$\mathcal{L}_{rank}(Q, D^+, D_1^-, \dots, D_N^-) = -\log \left(\frac{e^{sim(Q, D^+)}}{e^{sim(q, D^+)} + \sum_{i=1}^N e^{sim(q, D_i^-)}} \right)$$

where sim is the cosine similarity between representation of query documents Q and the document D .

Query generation: Query generation is the task of generating a short query, denoted as q , conditioned on the input text of query example documents, represented as Q . The goal of this task to effectively predicting the user’s intent and information needs. Following is the loss function of the query generation (qg) task:

$$\mathcal{L}_{qg}(q, Q) = -\sum_{t=1}^{|q|} \log P(q(t:t) | q(1:t-1), Q),$$

where $|q|$ denotes the length of query q and $q(j:k)$ represents the subquery extracted from q beginning at the j -th word and extending to the k -th word.

Total loss: The total loss is a weighted sum of the target task loss L_{rank} and the auxiliary task of L_{qg} defined as follows:

$$\mathcal{L}_{Total} = \alpha \times \mathcal{L}_{rank}(Q, D^+, D_1^-, \dots, D_N^-) + (1 - \alpha) \times \mathcal{L}_{pq}(q, Q),$$

where α is the weight hyper-parameter that balances the contribution of each loss term to the overall loss, allowing for the adjustment of the model’s focus between ranking accuracy and query generation quality.

System Details. We represent queries by concatenating all associated query documents and employ the Longformer Transformer architecture (Beltagy et al., 2020) for both query and candidate document encoding. Longformer’s extended context window (4096 tokens) accommodates our multi-document query representation. Input sequences are padded with [PAD] tokens if they fall short of the maximum context length. While our approach is adaptable to various Transformer architectures, we note that performance gains may be amplified when utilizing models with greater depth and dimensionality, at the cost of increased computation and memory requirements. For optimization, we use the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 3e-5 and train for a single epoch using a batch size of 32. We validate our method on the three query by example retrieval dataset Wiki-QBE dataset, Robust04-QBE and Multi-News-QBE.

Our methodology employs a standard two-stage ranking pipeline. For the first-stage retrieval, we adopt the Keyphrase method, as it demonstrated best performance in Section 5.2.1. To establish a baseline and isolate the impact of our auxiliary query generation task, we train a model variant, **Single-Loss-Rerank** where the query generation loss term \mathcal{L}_{qq} is masked. This allows us to compare with our proposed model, **Aux-Loss-Rerank**, which incorporates the auxiliary task during training.

Evaluation Metrics. For evaluating retrieval effectiveness at the reranking stage, we report mean average precision (MAP), mean reciprocal rank (MRR), and precision of the top 10 retrieved documents (P@10).

Results. Table 5.5 demonstrates the efficacy of the auxiliary training paradigm for query-by-example retrieval on the Wiki-QBE and Multi-News-QBE dataset. Compared to the first-pass retrieval (Keyphrase) and single-loss reranking (Single-Loss-Rerank), the auxiliary training approach (Aux-Loss-Rerank) yields the highest performance across all metrics for Wiki-QBE and Multi-News dataset. Both Single-Loss-Rerank and Aux-Loss-Rerank shows competitive performance to the Keyphrase

Table 5.5: Performance of Auxilary training paradigm for query by example retrieval task on Wiki-QBE dataset. Subscripts refer to the standard deviation of 5 corpuses. For Aux-Loss-Rerank, statistically significant improvements are marked by \star (over Keyphrase), \blacktriangle (over Single-Loss-Rerank).

Model	Wiki-QBE			Robust04-QBE			Multi-News-QBE		
	MAP	MRR	P@10	MAP	MRR	P@10	MAP	MRR	P@10
Keyphrase	0.1803 _(0.0050)	0.3970 _(0.0085)	0.1635 _(0.0038)	0.1371 _(0.0058)	0.5376 _(0.0202)	0.2957 _(0.0088)	0.4081 _(0.0032)	0.6090 _(0.0085)	0.1871 _(0.0005)
Single-Loss-Rerank	0.1895 _(0.0025)	0.4255 _(0.0109)	0.1768 _(0.0044)	0.1314 _(0.0041)	0.5192 _(0.0099)	0.2951 _(0.00970)	0.4118 _(0.0052)	0.6125 _(0.0104)	0.1932 _(0.0016)
Aux-Loss-Rerank	0.2110 \blacktriangle _(0.0064)	0.4617 \blacktriangle _(0.0125)	0.1898 \blacktriangle _(0.0061)	0.1372 _(0.0062)	0.5313 _(0.0105)	0.3095 _(0.0090)	0.4508 \blacktriangle _(0.0059)	0.6569 \blacktriangle _(0.0098)	0.2081 \blacktriangle _(0.0013)

model on Robust04-QBE. We hypothesize that the limited number of queries (≈ 200) used during fine-tuning is insufficient for the model to outperform the baseline on Robust04-QBE. This aligns with our observations of PRRIME variants’ performance on Robust04-QBE datasets in Section 5.3.

Table 5.6 shows the snippets of three example documents given as the query, the target query topic and the generated query topic by the decoder module. We observe that while the predicted query topic is not coherent in some places, it was able to capture the important information such as the place where the band was formed. This might be helpful for explainability by revealing the core focus of the final query representation. It is worth mentioning that by leveraging larger large language model architecture such as Llama2 we can improve upon the fluency of generated text.

5.6 Summary

In this chapter, we studied the task of query by example document where a user expresses an information need by providing single or multiple example documents. We focus on exploring how the contextual information embedded in query example documents could be effectively utilized to retrieve a ranked list of documents that align with the users’ information need. We leverage Transformer model architectures which are able to capture the context within a text using their attention mechanism

to represent query example documents and candidate documents in an information retrieval system.

First, we constructed three Query-By-Example (QBE) datasets: a large-scale dataset named Wiki-QBE for training purposes, and two evaluation datasets, Robust04-QBE and Multinews-QBE. We develop them by leveraging existing keyword-based datasets and based on the principle of that the query documents and the retrieval targets in the data collection are both relevant to the users’ information needs (*Contribution 3.1*).

We then introduced three end-to-end Transformer-based rerankers with the goal of improving the retrieval performance using the contextual information embedded in the query example documents:

- We developed the Passage-based Relevancy Representation with Multiple Examples (PRRIME) to overcome the limitations associated with the fixed-input sequence length of traditional Transformer models. This limitation becomes particularly challenging when dealing with long query and candidate documents. PRRIME addresses this by breaking documents into passages and utilizing an aggregated passage-level relevancy representation. Our findings demonstrate that PRRIME significantly improves upon the initial stage of retrieval, underscoring the effectiveness of its architecture in aggregating passage-level relevancy signals (*Contribution 3.2*).
- We developed a cross-encoder re-ranking architecture which uses a Transformer model for long inputs, namely Longformer. We demonstrated its effectiveness on the dataset that it is trained on by outperforming the first-stage retrieval. However, our results show that our model might be overfitted on the underlying characteristics of training corpus since it does not perform well on other datasets with limited number of instances. This indicates that while effective within its

training context, the model’s adaptability to diverse data conditions is limited. (*Contribution 3.3*).

- We introduced an auxiliary training framework employing a dual encoder ranking architecture, which integrates query generation as an auxiliary task. Our results indicate that this training method not only improves upon the initial stage of retrieval but also outperforms the single-loss ranking method. Additionally, the inclusion of precise short queries contributes to the model’s explainability, offering clear insights into the final query that is used for ranking (*Contribution 3.4*).

Our findings pave the way toward an effective query by example retrieval framework by utilizing the inherent capability of Transformer models to understand and encapsulate context.

Table 5.6

Query Example Documents	Target Query Topic	Predicted Query Topic
<p>Document1: Petrucci and her sister, Maxine Petrucci, first formed Madam X with vocalist Bret Kaiser and Chris Doliber. She left Madam X to join Vixen in 1986 and stayed until 1991. Roxy returned when Vixen reunited in 1997, bringing in her sister Maxine into the fold, but the lineup had to be dissolved the next year for legal reasons. ...</p> <p>Document2: She lives and works in Los Angeles, California and served from 2017 to 2019 as Chief Operating Officer, Executive Vice President of Business Affairs and Operations for The H Collective, a motion picture company. ...</p> <p>Document3: During its most commercially successful period from 1987 to 1992, the band consisted of Jan Kuehnemund lead guitar, Janet Gardner lead vocals, Share Ross bass guitar, and Roxy Petrucci drums. The band's eponymous first album was released in 1988, and reached No. ...</p>	<p>Vixen band. Vixen is an American rock band formed in Saint Paul, Minnesota, in 1980.</p>	<p>Vixen, the only half the size of the present, were the first to learn of the new information about the upcoming year's information about their progress. Vixen was formed in St. Paul, Minnesota, in the year of their first attempt to join the band. The band was led in part by Jule Kuehnemund, who was the first person to learn about it in the form of a book, and one of the first-year students.</p>

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this dissertation, we explored and enhanced tools and techniques for utilizing *contextual features* in representing queries and documents, aiming to improve the performance of information retrieval systems. Initially, we studied entities representing *things* in the real world and their relationships as a source of contextual information. Subsequently, we employed Transformer models, whose self-attention mechanisms excel at capturing contextual nuances in text, to build context-aware representations for various information retrieval tasks. Specifically, we examined the task of query expansion, using Transformer models to represent the context of pseudo-relevance feedback (PRF) documents for ranking terms relevant to the query. Additionally, we explored the task of query-by-example retrieval, where the user’s information need is latent in example documents. In this context, we leveraged Transformer models in various approaches to derive the query representation for an high-performance ranking system.

In Chapter 3, we introduced an entity embedding model designed to represent an entity by leveraging crucial related entities tagged in its summary. This embedding model was then employed in an entity ranking task, where both queries and documents were represented using this model. We integrated this embedding-based ranking model with a term-based ranking model, specifically Language Model retrieval, and demonstrated its enhanced performance over traditional term-based retrieval methods. Additionally, we evaluated its effectiveness against a word-based embedding model ranking approach, revealing that the entity-based embedding rank-

ing outperforms its counterpart. We also developed a fusion retrieval model that combines term-based language model retrieval, word-based embedding ranking, and entity-based embedding ranking, achieving the best performance. Furthermore, we applied the proposed entity-based expansion model to a query expansion task aimed at enhancing entity-centric complex queries retrieval. The expansion framework incorporates entities from both local and global knowledge sources: ‘local’ refers to entities indexed from the corpus, while ‘global’ pertains to those derived from the developed entity embedding model. Our experiments demonstrated that entity-based expansion outperforms all baseline word-expansion techniques available at the time of writing. Later works using Bert-based models (Nogueira & Cho, 2019) achieved higher performance.

In Chapter 4, we propose supervised and unsupervised methods using Transformer models for the task of query expansion. We build our methodology based on the robust query expansion method of relevance model which considers that the top- k retrieved documents in a first pass retrieval for the query q are relevant to q and can be used for obtaining relevant terms to the query for expansion. By leveraging Transformer models we are able to represent the context with these pseudo-relevant documents, as a result improving the ranking of relevant terms to query for expansion. Specifically, our unsupervised model, CEQE, utilizes BERT embeddings to calculate the similarity between a term in a PRF document and the query, taking into account the term’s context within that document. We demonstrate our unsupervised model, CEQE, is superior to the static embedding-based expansion models, and performs at least as well as state-of-the-art word-based feedback models on multiple collections. Additionally, our supervised model formulate query expansion as a **classification** task aiming to determine whether a term is relevant or non-relevant to the query. Our results show that the performance of SQET-variants is comparable to the term frequency based feedback models and the linear combination of the SQET model and

RM3, the term frequency based feedback model that we studied, results in further improvement comparing using them separately. Our research establishes a robust foundation for subsequent studies, such as those by Mackie et al. (2023b), which utilize Large Language Models (LLMs) to generate relevant text from various domains types—words, entities, questions, and more—in a zero-shot setting to enhance query expansion.

In Chapter 5, we study the task of query by example (QBE) retrieval and explore utilizing Transformer architecture to leverage the context within the query example documents for representing the latent query. We develop three QBE datasets: WikIR-QBE, a large-scale dataset for training, and two evaluation datasets, Robust04-QBE and Multinews-QBE. We propose three Transformer-based re-ranking architectures. Initially, we developed the PRRIME model, which addresses the limited context window of BERT-based Transformers. This model segments query example documents and candidate documents into passages, then trains an end-to-end neural ranking architecture that aggregates passage-level relevance representations. The results demonstrate an improvement of 37% and 19% over the first-stage term-only retrieval methods, which use the top- k ranked terms from TF-IDF as the query, on the Wiki-QBE and MultiNews-QBE datasets. Subsequently, we investigate the cross-encoder ranking architecture for the task of query by example retrieval employing the Longformer Transformer model, which was introduced for processing longer input texts at the time of studying this architecture. This approach shows high performance on the dataset it was trained on, Wiki-QBE, but struggles with transferability across other datasets – Robust04-QBE and MultiNews-QBE – likely due to overfitting on the text characteristics of the training set. Finally, we introduced a dual encoder architecture with an auxiliary query prediction task, which enhances both the first-stage retrieval phase and the performance of the dual encoder without the auxiliary task by 17% and 11% on Wiki-QBE and by 10% and 9% on MultiNews-QBE. These approaches

are among the first to leverage Transformer models for QBE retrieval, paving the way for future utilization of larger scale language models like GPT-3.5, GPT-4, Gemini, Llama, etc.

6.1 Future Work

In Chapter 3, we investigate entity-centric retrieval utilizing Word2vec-based embeddings. While recent research has explored Transformer architectures for this task 2.2, most approaches rely on unstructured text representations of entities, employing mean pooling to derive a single embedding representation. This may lead to the loss of granular information about entity attributes, hindering the retrieval of relevant results when queries focus on specific attributes. Preliminary studies (Gillick et al., 2019; Kong et al., 2022) have explored leveraging entity attributes to learn dense representations for improved retrieval, but further research is needed to develop models that effectively encode entities attributes to address nuanced user queries. This research direction holds promise not only for retrieving general entities from open-source knowledge graphs but also for enhancing search capabilities in diverse domains such as e-commerce, people search, and movie search.

In Chapter 4, we study query expansion utilizing BERT-based embedding vectors of terms in conjunction with pseudo-relevance feedback documents. The emergence of Large Language Models (LLMs) and their impressive zero-shot generation capabilities have motivated research into leveraging them for query reformulation (Mackie et al., 2023b), query intent identification (Mao et al., 2023), and utilizing LLMs’ generated response to the query for retrieval (Gao, Ma, Lin, & Callan, 2023; Jagerman, Zhuang, Qin, Wang, & Bendersky, 2023). However, existing studies have primarily focused on non-personalized domains. With the growing adoption of conversational systems across platforms like movie streaming and e-commerce, a promising research direction lies in employing LLMs for query expansion that incorporates user behavior and

interaction history by leveraging Retrieval Augmented Generation (RAG) approaches or fine-tuning.

In Chapter 5, we examine the task of query-by-example retrieval, briefly exploring the use of LLMs for zero-shot summarization and latent query representation, which yielded suboptimal results. However, multiple avenues exist to leverage LLMs to enhance query-by-example retrieval. For instance, given the challenge posed by long documents in this task, one potential research direction involves employing LLMs with extended context windows, fine-tuning them for embedding representation in a bi-encoder a dense retrieval approach.

BIBLIOGRAPHY

- Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., . . . Wade, C. (2004). Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, 189.
- Abolghasemi, A., Verberne, S., & Azzopardi, L. (2022). Improving bert-based query-by-document retrieval with multi-task optimization. In *Advances in information retrieval: 44th european conference on ir research, ecir 2022, stavanger, norway, april 10–14, 2022, proceedings, part ii* (pp. 3–12).
- Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., & Lin, J. (2019, November). Cross-domain modeling of sentence-level evidence for document retrieval. 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: Association for Computational Linguistics.
- Althammer, S., Hofstätter, S., Sertkan, M., Verberne, S., & Hanbury, A. (2022). PARM: A paragraph aggregation retrieval model for dense document-to-document retrieval. In M. Hagen et al. (Eds.), *Advances in information retrieval - 44th european conference on IR research, ECIR 2022, stavanger, norway, april 10-14, 2022, proceedings, part I* (Vol. 13185, pp. 19–34). Springer. Retrieved from https://doi.org/10.1007/978-3-030-99736-6_2 doi: 10.1007/978-3-030-99736-6_2
- Askari, A., & Verberne, S. (2021). Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieval. In *Proceedings of the second international conference on design of experimental search & information retrieval systems* (pp. 162–170).
- Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., . . . others (2016). Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.
- Balog, K., Carmel, D., & Arjen, P. (2012). de vries, daniel m. herzig, peter mika, haggai roitman, ralf schenkel, pavel serdyukov, thanh tran duc. In *The first joint international workshop on entity-oriented and semantic search (jiwes), acm sigir forum* (Vol. 46).
- Balog, K., & Neumayer, R. (2013). A test collection for entity search in dbpedia. In *Proceedings of the 36th international acm sigir conference on research and development in information retrieval* (pp. 737–740).
- Balog, K., Serdyukov, P., & Vries, A. P. (2010). *Overview of the TREC 2010 entity track* (Tech. Rep.). DTIC Document.
- Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

- Blanco, R., Halpin, H., Herzig, D. M., Mika, P., Pound, J., Thompson, H. S., & Duc, T. T. (2011). Entity search evaluation over structured web data. In *Proceedings of the 1st international workshop on entity-oriented search workshop (sigir 2011)*, acm, new york.
- Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.
- Boyotsov, L., Lin, T., Gao, F., Zhao, Y., Huang, J., & Nyberg, E. (2022). Understanding performance of long-document ranking models through comprehensive evaluation and leaderboarding. *arXiv preprint arXiv:2207.01262*.
- Breja, M., & Jain, S. K. (2021). A survey on non-factoid question answering systems. *International Journal of Computers and Applications*, 1–8.
- Caciularu, A., Cohan, A., Beltagy, I., Peters, M. E., Cattan, A., & Dagan, I. (2021). Cdlm: Cross-document language modeling. *arXiv preprint arXiv:2101.00406*.
- Cao, G., Nie, J. Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international acm sigir conference on research and development in information retrieval* (pp. 243–250). New York, NY, USA: ACM. doi: 10.1145/1390334.1390377
- Chatterjee, S., Mackie, I., & Dalton, J. (2024). Dreq: Document re-ranking using entity-based query understanding. In *European conference on information retrieval* (pp. 210–229).
- Cohan, A., Feldman, S., Beltagy, I., Downey, D., & Weld, D. S. (2020). Specter: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 2270–2282).
- Cohen, D., & Croft, W. B. (2016, September). End to end long short term memory networks for Non-Factoid question answering. In *Proceedings of the 2016 ACM international conference on the theory of information retrieval* (pp. 143–146). ACM.
- Cohen, D., Yang, L., & Croft, W. B. (2018). WikiPassageQA: A Benchmark Collection for Research on Non-factoid Answer Passage Retrieval. In *The 41st international ACM SIGIR conference on research & development in information retrieval, SIGIR 2018, ann arbor, mi, usa, july 08-12, 2018* (pp. 1165–1168). Retrieved from <https://doi.org/10.1145/3209978.3210118> doi: 10.1145/3209978.3210118
- Cortes, E. G., Woloszyn, V., Barone, D., Möller, S., & Vieira, R. (2021). A systematic review of question answering systems for non-factoid questions. *Journal of Intelligent Information Systems*, 1–28.
- Craswell, N., Mitra, B., Yilmaz, E., & Campos, D. (2019). Overview of the trec 2019 deep learning track. In *Proceedings of the twenty-eight text retrieval conference, TREC 2019, November 13-15, 2019*.
- Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search engines: Information retrieval in practice* (Vol. 520). Addison-Wesley Reading.

- Dai, Z., & Callan, J. (2019). Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 985–988).
- Dai, Z., & Callan, J. (2020a). Context-aware document term weighting for ad-hoc search. In *Proceedings of the web conference 2020* (pp. 1897–1907).
- Dai, Z., & Callan, J. (2020b). Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (pp. 1533–1536).
- Dalton, J., Dietz, L., & Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 365–374). New York, NY, USA: ACM.
- Dalton, J., Naseri, S., Dietz, L., & Allan, J. (2019). Local and global query expansion for hierarchical complex topics. In *European conference on information retrieval* (pp. 290–303).
- Demartini, G., Iofciu, T., & De Vries, A. P. (2009). Overview of the inex 2009 entity ranking track. In *International workshop of the initiative for the evaluation of xml retrieval* (pp. 254–264).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*. Minneapolis, Minnesota: Association for Computational Linguistics.
- Diaz, F., Mitra, B., & Craswell, N. (2016). Query expansion with locally-trained word embeddings. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*.
- Dietz, L., Gamari, B., & Dalton, J. (2018). *TREC CAR 2.1: A data set for complex answer retrieval*. Retrieved from <http://trec-car.cs.unh.edu>
- El-Arini, K., & Guestrin, C. (2011). Beyond keyword search: discovering relevant scientific literature. In *Proceedings of the 17th acm sigkdd international conference on knowledge discovery and data mining* (pp. 439–447).
- Fabbri, A. R., Li, I., She, T., Li, S., & Radev, D. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1074–1084).
- Ferragina, P., & Scaiella, U. (2012). Fast and accurate annotation of short texts with wikipedia pages. *IEEE software*, 29(1), 70–75.
- Foley, J., O’Connor, B., & Allan, J. (2016). Improving entity ranking for keyword queries. In *Proceedings of the 25th acm international on conference on information and knowledge management* (pp. 2061–2064).
- Formal, T., Piwowarski, B., & Clinchant, S. (2021). Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th international acm sigir conference on research and development in information retrieval* (pp. 2288–2292).

- Frej, J., Schwab, D., & Chevallet, J.-P. (2020, May). WIKIR: A python toolkit for building a large-scale Wikipedia-based English information retrieval dataset. In *Proceedings of the 12th language resources and evaluation conference* (pp. 1926–1933). Marseille, France: European Language Resources Association. Retrieved from <https://www.aclweb.org/anthology/2020.lrec-1.237>
- Fujii, A., Iwayama, M., & Kando, N. (2007). Overview of the patent retrieval task at the ntcir-6 workshop. In *Ntcir*.
- Gao, L., Dai, Z., Fan, Z., & Callan, J. (2020). Complementing lexical retrieval with semantic residual embedding. *arXiv preprint arXiv:2004.13969*.
- Gao, L., Ma, X., Lin, J., & Callan, J. (2023). Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1762–1777).
- Garigliotti, D., Hasibi, F., & Balog, K. (2019). Identifying and exploiting target entity type information for ad hoc entity retrieval. *Information Retrieval Journal*, 22, 285–323.
- Gerritse, E. J., Hasibi, F., & de Vries, A. P. (2020). Graph-embedding empowered entity retrieval. In *Advances in information retrieval: 42nd european conference on ir research, ecir 2020, lisbon, portugal, april 14–17, 2020, proceedings, part i 42* (pp. 97–110).
- Gerritse, E. J., Hasibi, F., & de Vries, A. P. (2022). Entity-aware transformers for entity search. In *Proceedings of the 45th international acm sigir conference on research and development in information retrieval* (pp. 1455–1465).
- Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldrige, J., Ie, E., & Garcia-Olano, D. (2019). Learning dense representations for entity retrieval. In *Proceedings of the 23rd conference on computational natural language learning (conll)* (pp. 528–537).
- Guo, J., Fan, Y., Ai, Q., & Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th acm international on conference on information and knowledge management* (pp. 55–64).
- Halpin, H., Herzig, D. M., Mika, P., Blanco, R., Pound, J., Thompon, H., & Tran, D. T. (2010). Evaluating ad-hoc object retrieval. In *Iwest@ iswc*.
- Hasibi, F., Balog, K., & Bratsberg, S. E. (2016). Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 acm international conference on the theory of information retrieval* (pp. 209–218).
- Hasibi, F., Nikolaev, F., Xiong, C., Balog, K., Bratsberg, S. E., Kotov, A., & Callan, J. (2017). Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval* (pp. 1265–1268).
- Henderson, M., Al-Rfou, R., Strobe, B., Sung, Y.-H., Lukács, L., Guo, R., ... Kurzweil, R. (2017). Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.
- Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., & Zhao, W. X. (2024). Large language models are zero-shot rankers for recommender systems. In *European conference on information retrieval* (pp. 364–381).

- Huang, Z., Naseri, S., Bonab, H., Sarwar, S. M., & Allan, J. (2023). Hierarchical transformer-based query by multiple documents. In *Proceedings of the 2023 acm sigir international conference on theory of information retrieval* (pp. 105–115).
- Hui, K., Yates, A., Berberich, K., & de Melo, G. (2017). A position-aware deep model for relevance matching in information retrieval. *arXiv preprint arXiv:1704.03940*.
- Hui, K., Yates, A., Berberich, K., & De Melo, G. (2018). Co-pacrr: A context-aware neural ir model for ad-hoc retrieval. In *Proceedings of the eleventh acm international conference on web search and data mining* (pp. 279–287).
- Imani, A., Vakili, A., Montazer, A., & Shakery, A. (2019). Deep neural networks for query expansion using word embeddings. In *European conference on information retrieval* (pp. 203–210).
- Jagerman, R., Zhuang, H., Qin, Z., Wang, X., & Bendersky, M. (2023). Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.
- Ju, J.-H., Yang, J.-H., & Wang, C.-J. (2021). Text-to-text multi-view learning for passage re-ranking. In *Proceedings of the 44th international acm sigir conference on research and development in information retrieval* (pp. 1803–1807).
- Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- Khattab, O., & Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (pp. 39–48).
- Kim, M.-Y., Rabelo, J., & Goebel, R. (2019). Statute law information retrieval and entailment. In *Proceedings of the seventeenth international conference on artificial intelligence and law* (pp. 283–289).
- Kim, Y., Rahimi, R., Bonab, H., & Allan, J. (2021). Query-driven segment selection for ranking long documents. In *Proceedings of the 30th acm international conference on information & knowledge management* (pp. 3147–3151).
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35, 22199–22213.
- Kong, W., Khadanga, S., Li, C., Gupta, S. K., Zhang, M., Xu, W., & Bendersky, M. (2022). Multi-aspect dense retrieval. In *Proceedings of the 28th acm sigkdd conference on knowledge discovery and data mining* (pp. 3178–3186).
- Kuzi, S., Shtok, A., & Kurland, O. (2016). Query expansion using word embeddings. In *Proceedings of the 25th acm international on conference on information and knowledge management*.

- Lavrenko, V., & Croft, W. B. (2001). Relevance based language models. In *Proceedings of the 24th annual international acm sigir conference on research and development in information retrieval* (pp. 120–127). New York, NY, USA: ACM. doi: 10.1145/383952.383972
- Lee, J., Fuxman, A., Zhao, B., & Lv, Y. (2015). Leveraging knowledge bases for contextual entity exploration. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1949–1958).
- Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3, 211–225.
- Li, C., Sun, Y., He, B., Wang, L., Hui, K., Yates, A., ... Xu, J. (2018). Nprf: A neural pseudo relevance feedback framework for ad-hoc information retrieval. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Li, C., Yates, A., MacAvaney, S., He, B., & Sun, Y. (2020). Parade: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093*.
- Lin, J. (2019). The neural hype and comparisons against weak baselines. In *Acm sigir forum* (Vol. 52, pp. 40–51).
- Lin, J., Nogueira, R., & Yates, A. (2021). Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4), 1–325.
- Lin, S.-C., Yang, J.-H., & Lin, J. (2020). Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386*.
- Lissandrini, M., Mottin, D., Palpanas, T., & Velegrakis, Y. (2018). Multi-example search in rich information graphs. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)* (pp. 809–820).
- Lissandrini, M., Mottin, D., Palpanas, T., & Velegrakis, Y. (2019). Example-based search: A new frontier for exploratory search. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (pp. 1411–1412).
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2023). Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.
- Liu, X., Chen, F., Fang, H., & Wang, M. (2014). Exploiting entity relationship for query expansion in enterprise search. *Information retrieval*, 17, 265–294.
- Liu, X., & Fang, H. (2015). Latent entity space: a novel retrieval approach for entity-bearing queries. *Inf. Retr. Journal*, 18(6), 473–503. Retrieved from <https://doi.org/10.1007/s10791-015-9267-x>
- Liu, Z., Xiong, C., Sun, M., & Liu, Z. (2018). Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. *arXiv preprint arXiv:1805.07591*.
- Lopez, V., Unger, C., Cimiano, P., & Motta, E. (2013). Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 21, 3–13.

- Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Lv, Y., & Zhai, C. (2009). 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* (pp. 1895–1898).
- Ma, X., Wang, L., Yang, N., Wei, F., & Lin, J. (2023). Fine-tuning llama for multi-stage text retrieval. *arXiv preprint arXiv:2310.08319*.
- Ma, X., Zhang, X., Pradeep, R., & Lin, J. (2023). Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.
- MacAvaney, S., Nardini, F. M., Perego, R., Tonellotto, N., Goharian, N., & Frieder, O. (2020). Expansion via prediction of importance with contextualization. *arXiv preprint arXiv:2004.14245*.
- MacAvaney, S., Yates, A., Cohan, A., & Goharian, N. (2019). CEDR: contextualized embeddings for document ranking. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, SIGIR 2019, paris, france, july 21-25, 2019*.
- MacAvaney, S., Yates, A., Cohan, A., Soldaini, L., Hui, K., Goharian, N., & Frieder, O. (2018, June). Characterizing question facets for complex answer retrieval. In *The 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 1205–1208). ACM.
- Mackie, I., Chatterjee, S., & Dalton, J. (2023a). Generative and pseudo-relevant feedback for sparse, dense and learned sparse retrieval. *arXiv preprint arXiv:2305.07477*.
- Mackie, I., Chatterjee, S., & Dalton, J. (2023b). Generative relevance feedback with large language models. In *Proceedings of the 46th international acm sigir conference on research and development in information retrieval* (pp. 2026–2031).
- Mao, K., Dou, Z., Mo, F., Hou, J., Chen, H., & Qian, H. (2023). Large language models know your contextual search intent: A prompting framework for conversational search. In *Findings of the association for computational linguistics: Emnlp 2023* (pp. 1211–1225).
- Metzler, D., & Croft, W. B. (2005). A Markov random field model for term dependencies. In *Proceedings of the 28th annual international acm sigir conference on research and development in information retrieval* (pp. 472–479). New York, NY, USA: ACM. Retrieved from <http://dx.doi.org/10.1145/1076034.1076115> doi: 10.1145/1076034.1076115
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In (Vol. 26).
- Mitra, B., Diaz, F., & Craswell, N. (2017). Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th international conference on world wide web* (pp. 1291–1299). Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. doi: 10.1145/3038912.3052579

- MontazerAlghaem, A., Zamani, H., & Allan, J. (2020). A reinforcement learning framework for relevance feedback. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (pp. 59–68).
- Mysore, S., O’Gorman, T., McCallum, A., & Zamani, H. (2021). Csfcube-a test collection of computer science research articles for faceted query by example. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*.
- Nanni, F., Mitra, B., Magnusson, M., & Dietz, L. (2017). Benchmark for complex answer retrieval. In *Proceedings of the acm sigir international conference on theory of information retrieval* (pp. 293–296).
- Naseri, S., Dalton, J., Yates, A., & Allan, J. (2021). Ceqe: Contextualized embeddings for query expansion. In *European conference on information retrieval* (pp. 467–482).
- Naseri, S., Dalton, J., Yates, A., & Allan, J. (2022). Ceqe to sqet: A study of contextualized embeddings for query expansion. *Information Retrieval Journal*, 25(2), 184–208.
- Naseri S., A. J., Foley J., & B., O. (2018). Exploring summary-expanded entity embeddings for entity retrieval [IR]. In *Ceur workshop proceedings*.
- Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., & Deng, L. (2016). Ms marco: A human generated machine reading comprehension dataset. In *Coco@ nips*.
- Ni, Y., Xu, Q. K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H. J., & Cao, S. S. (2016). Semantic documents relatedness using concept graph representation. In *Proceedings of the ninth acm international conference on web search and data mining* (pp. 635–644).
- Nogueira, R., & Cho, K. (2017). Task-oriented query reformulation with reinforcement learning. In *Emnlp*.
- Nogueira, R., & Cho, K. (2019). Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.
- Nogueira, R., Jiang, Z., & Lin, J. (2020). Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.
- Nogueira, R., Yang, W., Lin, J., & Cho, K. (2019). Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.
- Padaki, R., Dai, Z., & Callan, J. (2020). Rethinking query expansion for bert reranking. In *European conference on information retrieval*.
- Padigela, H., Zamani, H., & Croft, W. B. (2019). Investigating the successes and failures of bert for passage re-ranking. *arXiv preprint arXiv:1905.01758*.
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)*.
- Peters, M. E., Neumann, M., Logan IV, R. L., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

- Peters, M. E., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *Repl4nlp@acl*.
- Piroi, F., & Hanbury, A. (2019). Multilingual patent text retrieval evaluation: Clef-ip. In *Information retrieval evaluation in a changing world* (pp. 365–387). Springer.
- Poerner, N., Waltinger, U., & Schütze, H. (2020). E-bert: Efficient-yet-effective entity embeddings for bert. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 803–818).
- Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 275–281). New York, NY, USA: ACM.
- Pradeep, R., Sharifmoghaddam, S., & Lin, J. (2023). Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.
- Qiao, Y., Xiong, C., Liu, Z., & Liu, Z. (2019). Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.
- Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Shen, J., . . . others (2023). Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.
- Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W. X., . . . Wang, H. (2021). Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 5835–5847).
- Řehůřek, R., & Sojka, P. (2010, May 22). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). Valletta, Malta: ELRA. (<http://is.muni.cz/publication/884893/en>)
- Ristoski, P., & Paulheim, H. (2016). Rdf2vec: Rdf graph embeddings for data mining. In *International semantic web conference* (pp. 498–514).
- Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The smart retrieval system: Experiments in automatic document processing* (pp. 313–323). Prentice-Hall, Englewood Cliffs NJ.
- Roy, D., Paul, D., Mitra, M., & Garain, U. (2016, July 21). Using word embeddings for automatic query expansion.
- Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. In (Vol. 41, pp. 288–297). San Francisco, CA, USA: Wiley. Retrieved from <http://www.umiacs.umd.edu/~jimmylin/LBSC796-INFM718R-2006-Spring/papers/Salton90.pdf>
- Salton, G., Fox, E. A., & Wu, H. (1983). Extended boolean information retrieval. *Commun. ACM*, 26(11), 1022–1036.
- Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks* (pp. 92–101).

- Schuhmacher, M., Dietz, L., & Paolo Ponzetto, S. (2015). Ranking entities for web queries through text and knowledge. In *Proceedings of the 24th acm international on conference on information and knowledge management* (pp. 1461–1470).
- Schuster, M., & Nakajima, K. (2012). Japanese and korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Sekulić, I., Soleimani, A., Aliannejadi, M., & Crestani, F. (2020). Longformer for ms marco document re-ranking task. *arXiv preprint arXiv:2009.09392*.
- Serdyukov, P., & De Vries, A. (2009). *Delft university at the trec 2009 entity track: Ranking wikipedia entities* (Tech. Rep.). DELFT UNIV OF TECHNOLOGY (NETHERLANDS).
- Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., & Ma, S. (2020). Bertpli: Modeling paragraph-level interactions for legal case retrieval. In *Ijcai* (pp. 3501–3507).
- Shehata, D., Arabzadeh, N., & Clarke, C. L. (2022). Early stage sparse retrieval with entity linking. In *Proceedings of the 31st acm international conference on information & knowledge management* (pp. 4464–4469).
- Smucker, M. D., & Allan, J. (2006). Find-similar: similarity browsing as a search tool. In *Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval* (pp. 461–468).
- Song, H., Ren, Z., Liang, S., Li, P., Ma, J., & de Rijke, M. (2017). Summarizing answers in non-factoid community question-answering. In *Proceedings of the tenth acm international conference on web search and data mining* (pp. 405–414).
- Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1), 11–21.
- Sun, W., Yan, L., Ma, X., Ren, P., Yin, D., & Ren, Z. (2023). Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., . . . others (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Turtle, H., & Croft, W. B. (1991, July). Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst. Secur.*, 9(3), 187–222.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Vikraman, L., MontazerAlghaem, A., Hashemi, H., Croft, W. B., & Allan, J. (2021). Passage similarity and diversification in non-factoid question answering. In *Proceedings of the 2021 acm sigir international conference on theory of information retrieval* (pp. 271–280).
- Wang, Q., Kamps, J., Camps, G. R., Marx, M., Schuth, A., Theobald, M., . . . Mishra, A. (2012). Overview of the inex 2012 linked data track. In *Clef (online working notes/labs/workshop)*.

- Wang, X., MacAvaney, S., Macdonald, C., & Ounis, I. (2023a). Effective contrastive weighting for dense query expansion. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 12688–12704).
- Wang, X., MacAvaney, S., Macdonald, C., & Ounis, I. (2023b). Generative query reformulation for effective adhoc search. *arXiv preprint arXiv:2308.00415*.
- Wang, X., Macdonald, C., & Ounis, I. (2020). Deep reinforced query reformulation for information retrieval. *arXiv preprint arXiv:2007.07987*.
- Wang, X., Macdonald, C., Tonellotto, N., & Ounis, I. (2021). Pseudo-relevance feedback for multiple representation dense retrieval. In *Proceedings of the 2021 acm sigir international conference on theory of information retrieval* (pp. 297–306).
- Weng, L., Li, Z., Cai, R., Zhang, Y., Zhou, Y., Yang, L. T., & Zhang, L. (2011). Query by document via a decomposition-based two-level retrieval approach. In *Proceedings of the 34th international acm sigir conference on research and development in information retrieval* (pp. 505–514).
- Williams, K., Wu, J., & Giles, C. L. (2014). Simseerx: a similar document search engine. In *Proceedings of the 2014 acm symposium on document engineering* (pp. 143–146).
- Xiong, C., & Callan, J. (2015a). Esdrank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th acm international on conference on information and knowledge management* (pp. 951–960).
- Xiong, C., & Callan, J. (2015b). Query expansion with freebase. In *Proceedings of the 2015 international conference on the theory of information retrieval* (pp. 111–120). New York, NY, USA: ACM.
- Xiong, C., Callan, J., & Liu, T.-Y. (2016). Bag-of-entities representation for ranking. In *Proceedings of the 2016 acm international conference on the theory of information retrieval* (pp. 181–184).
- Xiong, C., Callan, J., & Liu, T.-Y. (2017). Word-entity duet representations for document ranking. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval* (pp. 763–772).
- Xiong, C., Dai, Z., Callan, J., Liu, Z., & Power, R. (2017). End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval* (pp. 55–64).
- Xiong, C., Liu, Z., Callan, J., & Hovy, E. (2017). Jointsem: Combining query entity linking and entity based document ranking. In *Proceedings of the 2017 acm on conference on information and knowledge management* (pp. 2391–2394).
- Xiong, C., Liu, Z., Callan, J., & Liu, T.-Y. (2018). Towards better text understanding and retrieval through kernel entity salience modeling. In *The 41st international acm sigir conference on research & development in information retrieval* (pp. 575–584).
- Xiong, C., Power, R., & Callan, J. (2017). Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web* (pp. 1271–1279).

- Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P., ... Overwijk, A. (2020). Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.
- Xiong, L., Xiong, C., Li, Y., Tang, K.-F., Liu, J., Bennett, P. N., ... Overwijk, A. (2021). Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International conference on learning representations*.
- Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In *In proceedings of the 19th annual international acm sigir conference on research and development in information retrieval* (pp. 4–11).
- Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., & Matsumoto, Y. (2018). Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. *arXiv preprint arXiv:1812.06280*.
- Yang, W., Xie, Y., Lin, A., Li, X., Tan, L., Xiong, K., ... Lin, J. (2019). End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics (demonstrations)*.
- Yang, Y., Bansal, N., Dakka, W., Ipeirotis, P., Koudas, N., & Papadias, D. (2009). Query by document. In *Proceedings of the second acm international conference on web search and data mining* (pp. 34–43).
- Yilmaz, Z. A., Wang, S., Yang, W., Zhang, H., & Lin, J. (2019). Applying bert to document retrieval with birch. 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): System demonstrations* (pp. 19–24).
- Yilmaz, Z. A., Yang, W., Zhang, H., & Lin, J. (2019). Cross-domain modeling of sentence-level evidence for document retrieval. 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)* (pp. 3490–3496).
- Yu, H., Xiong, C., & Callan, J. (2021). Improving query representations for dense retrieval with pseudo relevance feedback. In *Proceedings of the 30th acm international conference on information & knowledge management* (pp. 3592–3596).
- Yu, P., Huang, Z., Rahimi, R., & Allan, J. (2019). Corpus-based set expansion with lexical features and distributed representations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (pp. 1153–1156).
- Yu, P., Rahimi, R., Huang, Z., & Allan, J. (2020). Learning to rank entities for set expansion from unstructured data. In *Proceedings of the 2020 acm sigir on international conference on theory of information retrieval* (pp. 21–28).
- Zamani, H., & Croft, W. B. (2016). Embedding-based query language models. In *Proceedings of the 2016 acm international conference on the theory of information retrieval*.

- Zamani, H., & Croft, W. B. (2017). Relevance-based word embedding. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval* (pp. 505–514).
- Zerveas, G., Zhang, R., Kim, L., & Eickhoff, C. (2020). Brown university at trec deep learning 2019. *arXiv preprint arXiv:2009.04016*.
- Zhai, C., & Lafferty, J. (2001). 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: ACM.
- Zhan, J., Mao, J., Liu, Y., Zhang, M., & Ma, S. (2020). Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.
- Zhang, D., & Lee, W. S. (2009). Query-by-multiple-examples using support vector machines. *Journal of Digital Information Management*, 7(4), 202.
- Zhang, H., Song, X., Xiong, C., Rosset, C., Bennett, P. N., Craswell, N., & Tiwary, S. (2019). Generic intent representation in web search. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, SIGIR 2019, paris, france, july 21-25, 2019*.
- Zhao, J., Liu, M., Gao, L., Jin, Y., Du, L., Zhao, H., . . . Haffari, G. (2020). Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (pp. 1949–1952).
- Zheng, Z., Hui, K., He, B., Han, X., Sun, L., & Yates, A. (2020). Bert-qe: Contextualized query expansion for document re-ranking. In *Proceedings of the 2020 conference on empirical methods in natural language processing: Findings* (pp. 4718–4728).
- Zhiltsov, N., Kotov, A., & Nikolaev, F. (2015). Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In *Proceedings of the 38th international acm sigir conference on research and development in information retrieval* (pp. 253–262).
- Zhu, M., & Wu, Y.-F. B. (2014). Search by multiple examples. In *Proceedings of the 7th acm international conference on web search and data mining* (p. 667–672). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/2556195.2556206> doi: 10.1145/2556195.2556206
- Zhuang, H., Qin, Z., Hui, K., Wu, J., Yan, L., Wang, X., & Berdersky, M. (2023). Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. *arXiv preprint arXiv:2310.14122*.