Summer 2022

ANSWERING TOPICAL INFORMATION NEEDS USING NEURAL ENTITY-ORIENTED INFORMATION RETRIEVAL AND EXTRACTION

Shubham Chatterjee
University of New Hampshire

ANSWERING TOPICAL INFORMATION NEEDS USING NEURAL ENTITY-ORIENTED INFORMATION RETRIEVAL AND EXTRACTION

SHUBHAM CHATTERJEE

MS Computer Science, University of New Hampshire, Durham, 2020

DISSERTATION

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

in

Computer Science

September, 2022

This dissertation has been examined and approved in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science by:

Laura Dietz, Dissertation Director
Assistant Professor
Dept. Of Comp. Sc., UNH Durham

Marek Petrik
Assistant Professor
Dept. Of Comp. Sc., UNH Durham

Elizabeth Varki
Associate Professor/Graduate Program Coordinator
Dept. Of Comp. Sc., UNH Durham

Dongpeng Xu
Assistant Professor
Dept. Of Comp. Sc., UNH Durham

Jeff Dalton
Lecturer
Dept. Of Comp. Sc., University of Glasgow, UK

On May 17, 2022

Original approval signatures are on file with the University of New Hampshire Graduate School.

DEDICATION

To Maa, Baba, Rinku, Amma, and Debika for constantly supporting me through this journey. And to Maa Durga for never leaving my side. I couldn't have done it without them.

ACKNOWLEDGEMENTS

This dissertation has been made possible by generous financial support that I obtained from various sources throughout the course of my PhD. Any opinions, findings, and conclusions or recommendations expressed in this dissertation are those of the author and do not necessarily reflect the views of the financial bodies.

- A Teaching Assistant position at the Department of Computer Science, University of New Hampshire, Durham.
- The National Science Foundation under Grant No. 1846017 awarded to my advisor Dr. Laura Dietz.
- The Dissertation Year Fellowship provided by the Graduate School at the University of New Hampshire, Durham.

TABLE OF CONTENTS

ABSTRACT

ANSWERING TOPICAL INFORMATION NEEDS USING NEURAL ENTITY-ORIENTED INFORMATION RETRIEVAL AND EXTRACTION

Shubham Chatterjee

University of New Hampshire, September, 2022

In the modern world, search engines are an integral part of human lives. The field of Information Retrieval (IR) is concerned with finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need (query) from within large collections (usually stored on computers). The search engine then displays a ranked list of results relevant to our query. Traditional document retrieval algorithms match a query to a document using the overlap of words in both. However, the last decade has seen the focus shifting to leveraging the rich semantic information available in the form of *entities*.

Entities are uniquely identifiable objects or things such as places, events, diseases, etc. that exist in the real or fictional world. Entity-oriented search systems leverage the semantic information associated with entities (e.g., names, types, etc.) to better match documents to queries. Web search engines would provide better search results if they understand the meaning of a query.

This dissertation advances the state-of-the-art in IR by developing novel algorithms that understand text (query, document, question, sentence, etc.) at the semantic level. To this end, this dissertation aims to understand the fine-grained meaning of entities from the context in which the entities have been mentioned, for example, “oysters” in the context of food versus ecosystems. Further, we aim to automatically learn (vector) representations of entities that incorporate this fine-grained knowledge and knowledge about the query. This work refines the automatic understanding of text passages using deep learning, a modern artificial intelligence paradigm.

This dissertation utilized the semantic information extracted from entities to retrieve materials (text and entities) relevant to a query. The interplay between text and entities in the text is studied by addressing three related prediction problems: (1) Identify entities that are relevant for the query, (2) Understand an entity's meaning in the context of the query, and (3) Identify text passages that elaborate the connection between the query and an entity.

The research presented in this dissertation may be integrated into a larger system designed for answering complex topical queries such as *dark chocolate health benefits* which require the search engine to automatically understand the connections between the query and the relevant material, thus transforming the search engine into an answering engine.

PART I

INTRODUCTION

This page intentionally left blank.

CHAPTER 1

FOUNDATION: ENTITY-ORIENTED SEARCH

This dissertation advances the state-of-the-art in entity-oriented search. Before describing our goals and contributions in Chapter 2, we describe the terminology and foundation of entity-oriented search below.

Information retrieval. In the modern world, search engines are an integral part of human lives. We use Google, Bing, Baidu, etc. every moment as the main gateway to find information on the Web. The field of Information Retrieval (IR) is concerned with developing technology for matching *information needs* with *information objects*. According to Manning et al. [79],

Definition 1. *Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)*

Our information need (henceforth *query*), may range from a few simple keywords (e.g., *dark chocolate health benefits*) to a proper natural language question (e.g., *Who are the members of Eagle?*). The search engine then displays a ranked list of results, i.e., information objects relevant to our query. Traditionally, these items were documents. In fact, IR has been seen as synonymous with document retrieval by many.

Document retrieval models. Traditional document retrieval models such as BM25 [60], Language Models [107], and Term Frequency Inverse Document Frequency (TF-IDF) [23, 99, 118, 122, 128] are term based models that do not have any notion of semantics in them. For example, TF-IDF is a statistical measure used to evaluate the importance of a word to a document in a collection of documents (henceforth *corpus*). The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Similarly, BM25 is a *bag-of-words* retrieval function that represents text as the multi-set of its words, disregarding grammar, and even word order, but keeping multiplicity. BM25 ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document. Language Models are probability distributions over sequences of words where a separate language model is associated with each document in a corpus and documents are ranked based on the probability of the query Q in the document's language model M_d, i.e., $P(Q|M_d)$. None of these models consider the semantic relationship between various places, events, organizations, etc. in the query or the document.

A paradigm shift. However, there has been a dramatic shift in paradigm in the last decade with the focus shifting to leveraging the rich semantic information available in the form of *entities* [26, 54, 113, 144, 151]. Analysis of web search query logs and neural embedding models has shown that a large portion of the queries now contain some entity [53, 72, 110], reflecting an increase in the demands of users on retrieving relevant information about entities such as persons, organizations, products, etc. Advances in information extraction allow us to efficiently extract entities from free text [41, 83]. Since an entity is expected to capture the semantic content of documents and queries more accurately than a term, there has been much research in using entities to aid document retrieval and ranking.

1.1 What is an Entity?

An entity is a "thing" or "object" that one can refer to such as people, locations, diseases, events, etc. Identifying entities is an important and difficult task addressed by people in both the Natural Language Processing (NLP) as well as IR community (although traditionally, the task has been looked upon as more of a NLP problem than an IR problem). Balog [4] defines an entity as follows, taking inspiration from the Entity-Relationship (ER) Model proposed by Chen [20] in 1976:

Definition 2. *An entity is a uniquely identifiable object or thing, characterized by its name(s), type(s), attributes, and relationships to other entities.*

We restrict our universe to some particular registry of entities, which we will refer to as the *entity catalog*. Thus, we consider that an entity "exists" if an only if it is an entry in the given entity catalog. Thus:

Definition 3. *An entity catalog is collection of entries, where each entry is identified by a unique ID and contains the name(s) of the corresponding entity.*

1.2 Named Entities Versus Concepts

Often, entities are considered to be real-world objects. We may differentiate between two classes of entities:

1. **Named Entities.** These are the entities that exist in the real world such as people, locations, events, diseases, sports, etc.
2. **Concepts.** These are abstract objects such as mathematical concepts (e.g., theorem, distance, etc.), physical concepts (e.g., force, speed, etc.), psychological concepts (e.g., emotion, thought, etc.), or social concepts (e.g., peace, religion, etc.).

1.3 Properties of Entities

We refer to all the information associated with an entity as the *entity property*. The following are the most common entity properties:

- **Entity Identifier**. Each entity is associated with a unique identifier which helps to identify an entity. Examples of entity identifiers from past IR benchmarking campaigns include email addresses for people (within an organization), Wikipedia page IDs (within Wikipedia), and unique resource identifiers (URIs, within Linked Data repositories).

- **Name(s)**. Each entity is associated with a name. However, this name may not be unique. For example, the entity name *Apple* can refer to either the organization or the fruit. However, the ID associated with *Apple*, the organization is different from that of the fruit, which helps to disambiguate the entity references.

- **Type(s)**. Entities with similar properties are grouped together into a semantic type called an *entity type*. The set of possible entity types are often organized in a hierarchical structure, i.e., a *type taxonomy*. For example, the entity *Ed Sheeran* is an instance of the type “singer" which is a subtype of “person".

- **Attributes**. These are the characteristics or features of an entity. Each entity has different attributes. For example, a *person* entity might have attributes such as *date of birth*. *place of birth*, *name*, etc.

- **Relationships**. Relationships describe how two entities are associated to each other. For example, the entities *Barrack Obama* and *Michelle Obama* are related by the relation *is married to*.

1.4 Knowledge Graphs: Representing Properties of Entities

Consider the Wikipedia page of Barrack Obama. It contains information about him ranging from his early life, education, early career in law to his rise to US Presidency. Hence, to us humans, Wikipedia is a *Knowledge Repository*. According to Balog [4]:

Definition 4. *A Knowledge Repository is a catalog of entities that contains entity type information, and (optionally) descriptions or properties of entities, in a semi-structured or structured format.*

Wikipedia is a classic example of a knowledge repository. Each article in Wikipedia is an entry that describes a particular entity. Articles are also assigned to categories (which can be seen as entity types) and contain hyperlinks to other articles (thereby indicating the presence of a relationship between two entities, albeit not the type of the relationship).

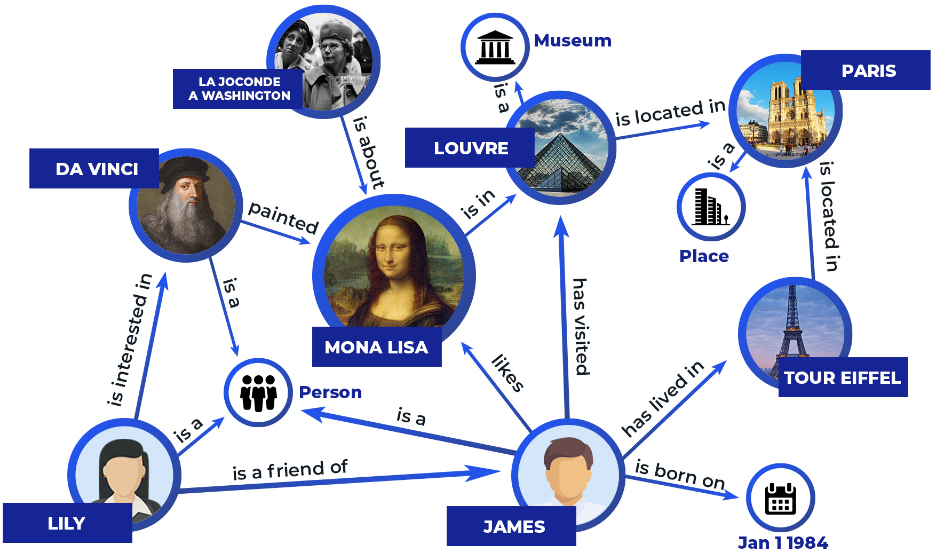


Figure 1.1: A Knowledge Graph. The nodes of this graph are entities and the edges are the relationships between these entities. For example, the entity "James" is related to the entity "Louvre" by the relationship *has visited*. Figure source: `https://yashuseth.wordpress.com/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/`

Wikipedia articles also contain information about attributes and relationships of entities, but not in a structured form.

With the development of knowledge repositories such as Wikipedia, a lot more information about entities have become available but for machines, this knowledge needs to be represented explicitly. A *Knowledge Base* (KB) is comprised of a large set of assertions about the world. To reflect how humans organize information, these assertions describe (specific) entities and their relationships. An AI system can then solve complex tasks, such as participating in a natural language conversation, by exploiting the KB. According to Balog [4]:

Definition 5. *A Knowledge Base is a structured knowledge repository that contains a set of facts (assertions) about entities.*

Conceptually, entities in a knowledge base may be seen as nodes of a graph, with the relationships between them as (labeled) edges. Thus, especially when this graph nature is emphasized, a knowledge base may also be referred to as a *Knowledge Graph* (KG) (Figure 1.1).

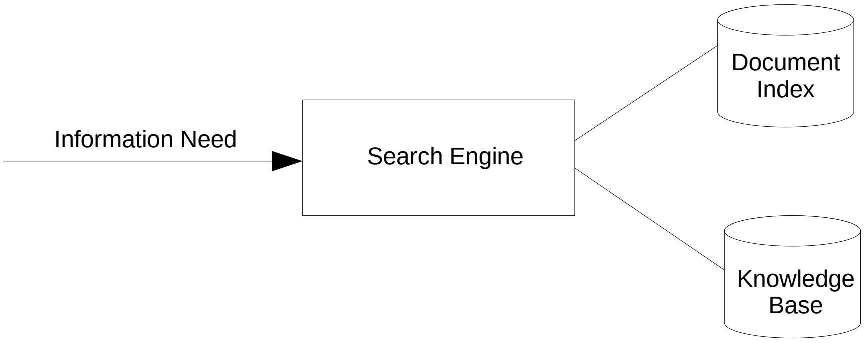


Figure 1.2: An entity-oriented search system. The search engine interacts with both, a document collection via a document index, and a knowledge base of entities.

1.5 Entity Linking

As described above, entities are associated with semantic information that is helpful for obtaining a deeper understanding of documents. Hence, for machines to understand documents, it is important that they are able to not only identify the entities in the document but also disambiguate them. For example, given a sentence such as “The apple fell on the ground”, machines must be able to identify that the word “apple” is actually an entity that refers to the fruit and not the company. We refer to this task as entity linking. According to Balog [4]:

Definition 6. *Entity linking is the task of recognizing entity mentions in text and linking them to the corresponding entries in a knowledge repository.*

Humans can accomplish this task easily due to their prior knowledge of the world but this task is extremely challenging for a machine. The availability of large knowledge repositories containing entities with unique ids has enabled the developement of entity linking systems that can identify and link an entity mention to an entity id.

1.6 Entity-Oriented Search

An entity-oriented search system (Figure 1.2) is similar to a conventional search system in that it interacts with a document collection via a document index. The difference lies in that an entity-oriented search system uses a Knowledge Base of entities to understand the document at a semantic level. The Knowledge Base contains the names of entities, the description of entities (often, the introductory paragraph from the Wikipedia article of the entity), and the type information of the entities (person, organization, etc.). This information is utilized by an entity-oriented search system to match a document to a user's information need.

CHAPTER 2

INTRODUCTION TO THIS DISSERTATION

Wikipedia is useful for users seeking information on topics such as “Genetically Modified Organism”; however, it is mostly focused on recent and popular topics only. The larger goal of this dissertation is to assist in the construction of systems that can (one day) answer broad topical information needs of users with a comprehensive Wikipedia-like article.

Users usually think of topics in terms of concepts or entities associated with the topic, the background stories and roles of these entities with respect to the topic, and how these entities are connected to each other and to the information needed. Entity-oriented search has become very popular in the last decade with some studies estimating that 40-70% of Web search queries target entities [53, 72, 110]. Motivated by this, we adopt an entity-oriented approach to identify relevant material (text and entities) for answering such topical queries. By identifying relevant material that may be included in a discussion on such a topic, we envision a downstream system to utilize these relevant materials to automatically construct a Wikipedia-like article for such topical queries.

With terminology of entity-oriented search introduced in Chapter 1, in this chapter, we introduce the research on entity-oriented information retrieval and extraction completed as part of this dissertation. First, we describe the motivation behind the research presented in this dissertation, then we briefly summarize the major contributions of this dissertation to the field of IR in general and entity-oriented search in particular.

Part of this chapter published as: Shubham Chatterjee. 2022. An Entity-Oriented Approach for Answering Topical Information Needs. In *Proceedings of the 44th European Conference on Information Retrieval (ECIR ’22).* Doctoral Consortium. Advances in Information Retrieval. Springer.

2.1 Motivation

Automatic algorithms for text understanding are essential for many artificial intelligence tasks. For example, in web search, search engines need to understand the text from the web pages. Such web pages are retrieved based on the overlap of words between the user's information need and the web page [60, 118, 122]. However, during the past decade, it has become popular to also use the rich semantic information available in the form of *entities* in text. These entities are stored in a semi-structured form in *knowledge bases*, along with some meta-information about each entity such as its name, type (person, location, etc.), and relationship to other entities in the knowledge base. The availability of large-scale knowledge bases has led to much research in using entities to aid document retrieval and ranking [26, 40, 76, 113, 144–147, 149, 152].

Below, we discuss the two major research directions of this dissertation that address two overarching issues in entity-oriented Web search to advance the state-of-the-art in the field.

2.1.1 Obtaining Fine-Grained Knowledge about Entities

A central question when leveraging entities for Web search is: How do we find entities described in queries and documents? This question has been extensively studied in the past, and tools (called *entity linking* tools) [41,83,109] which identify and disambiguate mentions of entities in text have been developed. Given some text such as *Oysters influence nutrient cycling and water filtration*, such tools can identify that the mention “Oyster" refers to the animal and not to the place[1] in Virginia, United States. However, it may be more beneficial for a user who is researching the role of oysters in ecosystems to know that the entity “oyster" has been mentioned in the context of its role as an ecosystem engineer and not its cultivation. Hence, there is a need for better tools which are able to understand text at a deeper level. Such fine-grained knowledge can aid downstream entity-oriented search and question answering systems which aim to understand the subtleties in human language.

[1] `https://en.wikipedia.org/wiki/Oyster,_Virginia`

Hence, a major focus of this dissertation is obtaining fine-grained understanding of an entity from the context in which the entity has been mentioned, and more specifically, in the context of a query. We aim to identify relevant text passages that elaborate on the connection between the query and entity. To this end, we use two sources for obtaining a *query-specific* description of an entity:

- **Wikipedia.** Wikipedia is a great source of knowledge about entities. Prior work in entity-oriented search has often used the introductory paragraph from the Wikipedia article of an entity as the entity's description. However, not all information on the Wikipedia page of an entity would be relevant to the entity in the context of the query. Hence, we identify the portion from the Wikipedia page that is helpful for understanding the meaning of the entity in the context of the query. To this end, we identify the relevant top-level sections from the Wikipedia page that best describes the entity in the given context. Following previous work [43,91,111,114], we refer to the top-level sections from Wikipedia as *aspects*, and use a catalog of aspects provided by Ramsdell et al. [111].[2] This aspect catalog contains the top-level sections from the entire English Wikipedia together with section heading, text of the section, and the entities mentioned in the section.

- **Entity-Support Passages.** The downside of using Wikipedia is that often, Wikipedia articles may be outdated or have some (negative) information removed. As a result, they do not contain all the query-relevant information. To alleviate the above problem, we explore an alternative source of query-specific entity descriptions. We use ideas from Pseudo-Relevance Feedback [70] to obtain an entity's query-specific description. Specifically, we identify pseudo-relevant passages that are relevant to the query and mention the entity in a salient i.e., central way and not as a passing reference. We refer to these passages as entity-support passages. We expect that entity-support passages obtained from a corpus can provide (query-specific) information which is complimentary to the (missing) information on the Wikipedia page of an entity.

[2] https://www.cs.unh.edu/~dietz/eal-dataset-2020/

Recently, the entity-support passage retrieval task (in various flavors) has also received much attention from the research community. For example, the entity retrieval task of the Complex Answer Retrieval track [33] at the Text Retrieval Conference (TREC) is to retrieve Wikipedia entities in response to a query, along with passages from Wikipedia which explain how the entity is related to the query. Similarly, the goal of the *Wikification* task at the TREC News [127][3] track is to link the entities in news articles to an external resource such as Wikipedia which provides more information on the linked entity. The Retrieval From Conversational Dialogues (RCD)[4] track at the Forum for Information Retrieval (FIRE) 2020 had a passage retrieval task where given an excerpt of a dialogue, the task is to return a ranked list of passages from Wikipedia which contain information on the entities in the dialogue.

2.1.2 Query-Specific Entity Representations

Another central question that one often encounters in entity-oriented research is: How do we represent the entities so that they are understandable by a machine? As machines only understand numbers, the common approach is to use the vector representation of the introductory paragraph from the Wikipedia page of an entity [77, 80, 151]. This vector representation may be obtained, for example, using TF-IDF, where each entry in the vector is the TF-IDF of the word corresponding to that entry. The issue with this approach is that the introductory paragraph is often a generic description of the entity without any knowledge of the query.

IR systems deal with explicit queries that are not known beforehand. Often, queries and documents are matched [77, 80, 91, 151] through the (cosine) similarity between the (vector) representation of the entities mentioned in the query and the document. Entity representations without any knowledge of the query may not be able to identify when two entities are similar/related in the context of the query. For example, the Wikipedia page of the entity "Food and Drug Administration" does not mention the entity "Robert Swanson", yet these two entities are similar/related in the context of the query "Genetically Modified

[3] http://trec-news.org/

[4] https://rcd2020firetask.github.io/RCD2020FIRETASK/

Organism" because Robert Swanson was the founder of the company that produced the first genetically engineered insulin approved for use by the Food and Drug Administration. As a result, the representation of entities obtained via the introductory paragraph from Wikipedia or even large-scale Knowledge Graphs [14,73,121,139] may not be suitable for IR systems. Hence, another major focus of this dissertation is automatically learning entity representations, and more specifically, query-specific entity representations.

2.2 Use Cases

We study the utility of entity-support passages/entity aspects/query-specific entity representations for two downstream tasks, one where the query is short (as is typical in IR), and the other where the query is longer. This provides us a way to generalize the efficacy of our approaches and also study the various error modes that our approach makes under different conditions. The two tasks/use cases are described below.

- **Entity retrieval.** Given a query, find relevant entities for the query. In this task, the query is short (e.g., question, keywords). We learn query-specific entity representations using entity-support passages and entity aspects, and leverage these query-specific entity representations for entity retrieval. Further, we also study whether entity-support passages and entity aspects can help an entity retrieval system when used directly (not for learning representations).

- **Entity aspect linking.** Given a set of pre-defined entity aspects (top-level Wikipedia sections), find the aspect that best matches the provided entity context (e.g., paragraph). Here, the "query" is long (e.g., a text passage): The context may be considered as the "query" and the aspect as the "document" to be matched. Here, we study the utility of learning context-dependent (query-specific) entity representations for matching two long-form texts within a neural network framework, when large-scale training data is available. Furthermore, we study whether entity aspect linking can benefit by ranking the entities contained in the aspect using the context as a query. As a side contribution, we also study whether current entity salience detection systems are enough to reap the benefits of entity salience for entity aspect linking: Intuitively, an aspect should be similar to the context that mentions an entity found to be salient in the aspect.

2.3 Contributions

This dissertation summarizes novel insights into entity-oriented information retrieval and extraction. While we demonstrate significant performance improvements for information retrieval tasks, we anticipate a positive impact on many related fields, such as Natural language Processing and Semantic Web. Below, we summarize the major findings/contributions of this dissertation which are further detailed in Chapters 5 through 8. Note, the contributions below are the "big" findings that affect the field of IR in general. More concrete contributions are included with the respective chapters.

1. **Query-specific entity representations.** While language models such as BERT (for details see Chapter 4.5) have been shown to be useful for the document ranking task [80, 93–95], we find that BERT does not understand entities very well. Ideally, given the name of an entity, a neural language model should be able to generate a representation of that entity that not only encodes the general knowledge about the entity available in a Knowledge Graph but also query-specific knowledge about the entity. However, we find that this is not the case. As a result, the application of BERT in entity-oriented search is limited. We show that it is possible to inject query-specific knowledge into BERT to learn query-specific entity representations that are ultimately useful for a downstream IR task. This is achieved using entity-support passages and entity aspects. Through this dissertation, we make a significant contribution to the emerging and growing field of learning BERT-based entity representations [103, 106, 138, 159]: Our BERT entity representations are query-specific whereas all prior BERT entity representations do not incorporate the query. We study the utility of entity representations in the context of the entity ranking task, and show that query-independent entity representations are not ideal for IR tasks; significant performance improvements can be obtained when using our query-specific BERT entity

representations.

2. **Entity descriptions.** We show that it is important to consider the query while selecting a description of an entity. Prior work on entity-oriented search often uses the introductory paragraph from the Wikipedia page of an entity as the entity's description. While this approach is easy-to-implement, we show that it is not sufficient to do well on IR tasks, and that better results can be obtained when using our query-specific entity descriptions. In this regard, we repeatedly find that Pseudo-Relevance Feedback (PRF) is helpful for obtaining a query-specific entity description. As such, PRF forms the basis of a lot of work presented in this dissertation.

3. **Entity retrieval.** Entity retrieval is a very important task in IR: Often, the information need of Web searches can be answered using a single entity (such as in conversational retrieval or factoid question answering), or a list of entities (such as for the query *Who are the members of the Beetle?*). The research presented in this dissertation significantly advances the state-of-the-art in entity retrieval by leveraging entity-support passages and entity aspects for entity retrieval.

4. **Entity frequency is a strong indicator.** We find the a "simple" entity statistic such as frequency is a very strong indicator of relevance and can significantly improve retrieval performance. In particular, we find that frequency of co-occurrence of entities correlates strongly with the relatedness between the entities. This is shown in our work on entity aspect linking (Chapter 8). In our work on entity-support passage retrieval (Chapter 5), we find that identifying passages that contain entities frequently co-occurring with the target entity (the entity for which we want to find the support passage) is a good support passage for the target entity. Later, in Chapters 6 and 7, we find that entity-support passages identified using frequently co-occurring entities are better entity descriptions and helpful for learning query-specific entity representations. These entity representations are shown to improve performance on the entity retrieval task as compared to several baselines on two large-scale entity ranking test collections.

5. **Entity salience.** Entity salience has been well-studied in the NLP community; however, its utility for IR tasks is not very well-studied. Intuitively, a text passage that mentions a relevant entity in a salient way must also be relevant for the query. Through this dissertation, we study whether indicators of entity salience are useful for finding when a text passage is relevant to a query. We study the utility of entity salience for IR in the context of entity aspect linking and entity-support passage retrieval. Our findings with respect to entity salience for IR is detailed in Chapters 5 and 8.

2.4 Publications

Below, we list all the research papers that were produced from research completed as part of this dissertation and published at top peer-reviewed conferences in the field of IR.

1. **Shubham Chatterjee** and Laura Dietz. 2019. ***Why Does This Entity Matter? Support Passage Retrieval for Entity Retrieval.*** In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '19)*. Association for Computing Machinery, New York, NY, USA, 221–224. In this work, we address the entity-support passage retrieval task and explore the utility of entity salience for the task. This work is detailed in Chapter 5.

2. **Shubham Chatterjee** and Laura Dietz. 2021. ***Entity Retrieval Using Fine-Grained Entity Aspects.*** In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, New York, NY, USA. In this work, we leverage entity aspects and entity-support passages for entity retrieval and achieve significant performance improvements over several baselines. This work is detailed in Chapter 6.

3. **Shubham Chatterjee** and Laura Dietz. 2022. ***BERT-ER: Query-specific BERT Entity Representations for Entity Ranking.*** In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. Association for Computing Machinery, New York, NY, USA. In this work,

we utilize entity aspects and entity-support passages to automatically learn query-specific vector representations of entities using BERT. This work is detailed in Chapter 7.

4. **Shubham Chatterjee**. 2022. ***An Entity-Oriented Approach for Answering Topical Information Needs.*** In *Proceedings of the 44th European Conference on Information Retrieval (ECIR '22).* Springer-Verlag, Berlin, Heidelberg. This work is an abridged version of this dissertation presented at the Doctoral Consortium in ECIR 2022 in Norway.

5. Laura Dietz, **Shubham Chatterjee**, Connor Lennox, Sumanta Kashyapi, Pooja Oza, and Ben Gamari. 2022. ***Wikimarks: Harvesting Relevance Benchmarks from Wikipedia.*** In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22).* Association for Computing Machinery, New York, NY, USA. As this dissertation uses Wikipedia to a great extent, several entity ranking and passage ranking baselines were contributed by the author of this dissertation for release with this dataset.

2.5 Dissertation Outline

The remainder of this dissertation is organized as follows. In Chapter 3, we provide a brief survey of the current state-of-the-art in various topics related to research presented in this dissertation. In Chapter 4, we provide a high-level overview of necessary concepts from deep learning for NLP that are required for understanding the research presented in this dissertation. Chapters 5 through 7 describe our novel research in advancing the state-of-the-art in entity retrieval. In particular, Chapter 5 discusses our research on entity-support passage retrieval, Chapter 6 discusses our work on leveraging fine-grained knowledge obtained using entity aspects to improve entity retrieval performance, and Chapter 7 discusses our work on learning query-specific entity representations and its utility for entity retrieval. Chapter 8 explores the utility of entity salience and entity relatedness for the entity aspect linking task. Finally, we conclude the dissertation with Chapter 9.

CHAPTER 3

RELATED WORK

3.1 Entity-Support Passage Retrieval

Given a query and a target entity, the entity-support passage retrieval task is to find a paragraph-size text from a corpus that elaobrates on the connection between the query and entity by explaining why/how the entity is relevant to the query. In addition, the entity must be salient, i.e., *central* to the discussion in the support passage.

3.1.1 Connecting Entities to Queries

Blanco et al. [13] present a model that ranks entity support sentences with learning-to-rank. They present several retrieval-based, entity-based and position-based methods and use features based on named entity recognition (NER) in combination with term-based retrieval models. Their approach consists of first segmenting the document into sentences and using a sentence-entity matrix to represent the presence of an entity in the sentence. They frame the problem as a ranking problem for triples of *(sentence, query, entity)*. Several ranking features are employed, such as the BM25 retrieval score of the sentences, the original retrieval score of the entity for the query, the distance between the last match of query and entity, and the length of the sentence, etc. We include this work as a baseline in our work.

Kadry et al. [61] study whether relation extraction is beneficial for support passage retrieval, and the limitations of the current relation extraction approaches that need to be overcome. As such, most of their features are based on relation-extraction and Natural Language Processing. These features are then used in a learning-to-rank framework.

3.1.2 Connecting Entities to Entities

A task related to entity-support passage retrieval is *entity relation explanation*. Given a pair of entities in a knowledge graph, the entity relation explanation task is to find a passage which explains the relationship of these two entities in the knowledge graph. One approach in literature is to treat the problem from a graph perspective. For example, Pirro et al. [105] consider the problem as a sub-graph finding problem, where the sub-graph consists of nodes and edges in the set of paths between the two input entities, whereas Aggarwal et al. [1] rank all the paths between any two entities in a knowledge graph. Voskarides et al. [135, 136] use textual, entity and relationship features within a LTR framework, whereas Bhatia et al. [10] address the problem from a probabilistic perspective.

3.2 Entity Retrieval

Given a query (question, keyword, etc.), the entity retrieval task is to retrieve entities from a Knowledge Graph ordered by relevance of each entity to the query.

3.2.1 Term-Based Models for Entity Ranking

The first step in entity retrieval is often the construction of a term-based representation of entities called *entity description*. Such entity descriptions can be created from semi-structured documents such as Wikipedia by representing the entities using a *fielded document*, where fields in the document correspond to specific parts of Wikipedia such as title, introductory text, entity links, etc [26, 80, 81]. Alternatively, structured data about entities in the form of subject-predicate-object (SPO) triples that are available in large-scale knowledge bases such as DBpedia can be converted to term-based representation [19, 52, 63]. For entities which do not have a ready-made entity description available, such as entities found in a web crawl, entity descriptions may be constructed by considering the mentions of the given entity across the document collection [5, 6].

Ranking via unstructured retrieval models. Once an entity description has been constructed, traditional retrieval models such as BM25 [116] and Language Models may be employed to rank these descriptions (and hence entities).

However, models such as those above do not consider term dependencies. Markov Random Fields (MRF) [85] are used to model term dependence. For example, the Sequential Dependence Model (SDM) [85] is based on the MRF and assigns different weights to matching query term concepts (e.g., unigram, bigrams, etc.) of different type. The Weighted Sequential Dependence Model [9], a variant of the SDM estimates the importance of each matching query term concept individually. Raviv et al. [112] model the different representations of an entity (description, type, and name) jointly with the query terms using MRFs.

Ranking via fielded retrieval models. An entity description can be created from semi-structured documents about entities on the Web, such as the Wikipedia page of the entity, by representing the entity as a *fielded document*. Each field in the document corresponds to specific parts of the Wikipedia page, such as title, introductory text, entity links, etc [26, 26, 80, 81]. Several extensions of the models described above consider multiple fields.

BM25F [117], the fielded variant of BM25, uses weighted term frequencies calculated as a linear combination of term frequencies across the different fields, with field-specific normalization applied.

The Mixture of Language Models (MLM) [96] estimates a separate language model for each entity field, and then combines these field-level language models into an entity-level model using a linear mixture. The linear mixture model learns a weight w_f for each entity field using training data and the Coordinate Ascent algorithm [84]. Alternatively, w_f may also be set using the Probabilistic Retrieval Model for Semi-structured Data (PRMS) proposed by Kim et al. [67]. PRMS replaces the static field weight w_f with the probability of mapping a term t from the entity description to the given entity field f, referred to as the mapping probability, $P(f \mid t)$. $P(f \mid t)$ is estimated using Bayes' Theorem, where the probability of a term given a field, $P(t \mid f)$, is estimated using the background language model of that field, and $P(f)$ is either left uniform or used to incorporate domain-specific knowledge.

Zhiltsov et al. [160] propose the Fielded Sequential Dependence Model (FSDM) that combines the SDM and MLM. FSDM estimates the feature functions for unigams and bigrams across multiple fields using field-specific background models. However, the FSDM has two limitations.

First, even with a few fields, the FSDM has to estimate a large number of free parameters λ and field weights w_f. To over come this issue, Hasibi et al. [54] propose to estimate the field weights w_f using field mapping probability estimates $P(f \mid t)$ from PRMS.

Second, the field weights w_f are the same for all query term concepts of the same type (unigrams, ordered and unordered bigrams), which can be a problem if any query concept is incorrectly projected onto a different field. Nikolaev et al. [92] propose two variations of the FSDM to overcome this issue: Paramterized Fielded Sequential Dependence Model (PFSDM) and Parametrized Fielded Full Dependence Model (PFFDM). These models dynamically estimate the probability of a query concept being mapped onto a field using some statistical and linguistic features.

Ranking via Learning-To-Rank. Learning-To-Rank (LTR) approaches effectively combine a large number of entity relevance indicators from multiple sources. Each query-entity pair is represented as a feature vector, and the optimal way to combine these vectors is learnt through discriminative training. Hence, the performance is highly dependent on the choice of features and the amount of training data available.

The most commonly used features are *query-entity* features which capture the degree of similarity between the query and the entity. Schuhmacher et al. [123] propose several query-entity features, for example, whether the candidate entity is contained in the query entity, whether the entities in the query and those in the document have an edge in a Knowledge Graph, etc.

Graus et al. [52] propose a LTR approach based on fielded document representation of entities. However, in contrast to the previous approach of representing entities as fielded documents with information from the *same* source, Graus et al. use content from *different* sources (Knowledge Base, Twitter, etc.) in each field of the document. Then, an optimal entity representation for entity retrieval is learnt using three types of features: (1) Field similarity features which model the query-field similarity, (2) Field importance features which favor fields with more and novel content, and (3) Entity importance features which aim to favor recently updated entities.

Dietz [32] proposed ENT Rank, a LTR model which utilizes the information about text for entity retrieval. Neighbor relations between entities are defined using the context of the entity such as the passage in which the entity appears.

Yamada et al. [155] proposed Wikipedia2Vec for learning embeddings of words and entities from Wikipedia based on the skip-gram model [86, 87]. Gerritse et al. [50] propose GEEER, an entity ranking system that re-ranks entities using Wikipedia2Vec. They show that entity embeddings from Wikipedia2Vec are useful for entity ranking. First, they calculate the embedding-based score for an entity E as follows:

$$\text{Score}_{\text{emb}}(E, Q) = \sum_{e \in Q} C(e) \cdot \cos(\vec{E}, \vec{e}) \tag{3.1}$$

where $C(e)$ is the confidence score of each entity e in the query Q as returned by the entity linker TagMe [42]. The final score of an entity is:

$$\text{Score}_{\text{final}}(E, Q) = \lambda \cdot \text{Score}_{\text{emb}}(E, Q) + (1 - \lambda) \cdot \text{Score}_{\text{ret}}(E, Q)$$

where $\text{Score}_{\text{ret}}(E, Q)$ is the score of entity E obtained from a retrieval model, and $\lambda \in [0, 1]$.

3.2.2 Semantically Enriched Models for Entity Ranking

Several models have been proposed in literature which leverage entity-specific properties such as attributes, types and relationships available in large-scale Knowledge Bases to improve entity retrieval. Such models enrich the query with the semantic representations of the entities in the query for the purpose of matching queries and entities. Below, we give a brief overview of such models.

Ranking via entity types. Models which utilize entity types for entity ranking use an enriched version of the query q' consisting of the keyword query q, and the set of target types T_q. The target types T_q may be provided by the user [6] or identified from a Knowledge Base [30, 49]. The general formulation of a type-aware scoring function consists of a linear mixture of two components: (1) Score(e, q), the term-based similarity between an entity e and the keyword query q, and (2) Score(e, T_q), the type-based similarity between the entity e and the set of target types T_q. There are several choices for estimating Score(e, T_q).

One may use a fielded representation of the entity, with a separate field for the labels of the corresponding entity types. The similarity between the labels of the target types and this field can then be estimated using any term-based retrieval model described above [29]. Alternatively, types may also be represented by concatenating the descriptions of entities that belong to that type, and then scoring the query against this representation [62]. Pehcevski et al. [100] propose a set-based similarity between set of target types T_q and the set of types of an entity T_e by measuring the ratio of common types between T_q and T_e. Raviv et al. [112] measure the type-based similarity using the distance of the entity and the query types in the taxonomy. Balog et al. [6] represent query and entity types using probability distributions, and then measure the similarity between the two distributions.

Ranking via entity relationships. Entities in a Knowledge Graph are connected via edges which represents how these entities are related to each other. Often, queries can be answered by leveraging these entity relationships. For example, Tonnon et al. [131] address the ad-hoc entity retrieval task by first identifying a set of seed entities using term-based retrieval, and then traversing the edges of these entities in the Knowledge Graph to identify other related entities (relevant) entities.

Ciglan et al. [22] propose the *SemSets* model to address the List Search task of the Semantic Search Challenge [11]. *SemSets* consists of three steps: (1) Identify and score possible candidate entities to answer the query, (2) Identify sets of semantically related entities from the underlying Knowledge Graph and score entities based on the relevance score of sets it belongs to, and (3) Identify the principal entity in the query and score an entity based on its distance from this principal entity.

Bron et al. [15] address the Related Entity Finding task of the TREC Entity track [7]. The input is an enriched query q' consisting of the keyword query q describing the relation, the input entity e_q, and the target type y_q. The relevance of an entity e is modelled using a generative probabilistic model in three steps: (1) Score the entity e based on the strength of its association to the input entity e_q, (2) Estimate the probability that e is of the target type y_q, and (3) Estimate the likelihood that the relation contained in the keyword query q is found around the pair (e, e_q).

3.3 Learning Entity Representations

The large-scale nature and sparsity of Knowledge Graphs (KGs) requires that we represent the KGs using distributed representations. As a result, there has been much research in the area of embedding both entities and relations from a KG into a continuous low-dimensional space. Below, we briefly outline the major works in this area.

3.3.1 Graph Embeddings

Bordes et al. [14] propose TransE which learns embeddings for both entities and relations based on the idea that the relationship r between two entities h and t corresponds to a translation between the embedding of these entities. However, TransE has problems dealing with reflexive, one-to-many, many-to-one, or many-to-many relations between entities. Wang et al. [139] propose TransH to overcome this issue by representing each relation r with two vectors: the norm vector $\mathbf{w}_r$, and the translation vector $\mathbf{d}_r$.

Both TransE and TransH assume that entities and relations are embedded in the same space. Lin et al. [73] propose TransR to address this issue by modelling entities and relations in distinct entity space and multiple relation spaces. TransR projects $\mathbf{h}$ and $\mathbf{t}$ to the aspects that a relation r focuses on using relation-specific mapping matrix $\mathbf{M}_r$. However, this means that for relation r, all entities share the same $\mathbf{M}_r$ irrespective of their types or attributes. Ji et al. [59] propose TransD to address this issue by using a unique mapping matrix for every entity-relation pair.

Xie et al. [142] propose a novel representation learning method for knowledge graphs taking advantages of entity descriptions present in knowledge bases. Yamada et al. [157] present *TextEnt*, a neural network model that learns distributed representations of entities and documents directly from a knowledge base using the introductory paragraph of an entity's Wikipedia article as descriptions.

3.3.2 Knowledge-Enhanced BERT

Recently, much effort has been spent on injecting knowledge into BERT [31]. Zhang et al. [159] propose ERNIE, a neural language model that uses additional knowledge encoder layers to integrate the knowledge from entities into the textual information from the underlaying layers. Peters et al. [103] propose KnowBert, a knowledge-enhanced BERT model that explicitly models entity spans in the input text and uses an entity linker trained jointly with the model to retrieve relevant entity embeddings. Using a word-to-entity attention, KnowBert allows long range interactions between contextual word representations and all entity spans in the context. Wang et al. [138] propose KEPLER, a RoBERTa-based model that maps texts and entities onto the same semantic space using the same language model and jointly optimizes the Knowledge Embedding and the Masked Language Modeling objectives. While ERNIE and KnowBert are based on adapting BERT to entity embeddings and involve additional pre-training, E-BERT proposed by Poerner et al. [106] adapts entity embeddings to BERT without any pre-training. E-BERT aligns Wikipedia2Vec entity vectors with BERT's word-piece vectors. E-BERT has been shown to outperform BERT, ERNIE, and KnowBert on question-answering, relation classification, and entity linking.

3.4 Entity Aspect Linking

Given an entity mention in a context such as a sentence, tweet or paragraph, and a set of predefined aspects along with the associated content (text and entities), the entity aspect linking task is to find the aspect from the set that best captures the topic addressed in the context.

3.4.1 Sections as Entity-Aspects

Fetahu et al. [43] were the first to define Wikipedia sections as entity aspects. Although the paper does not explicitly refer to sections as “aspects”, it considers each section as a separate sub-topic of the entity. They enrich Wikipedia sections with news-article references in two steps: First, they suggest news articles to Wikipedia entities (article-entity placement step) and Second, they find the exact section in the entity page where the article must be placed (article-section placement step).

Similarly, Banerjee et al. [8] seek to improve Wikipedia stubs by generating content for each section automatically. Their system is based on a text classifier which uses topic distribution vectors to assign content from the web to various sections on a Wikipedia article. This is followed by an abstractive summarization step where section-specific summaries for Wikipedia stubs are generated.

Following these works, Reinanda et al. [114] define an entity’s aspects as the top-level sections from Wikipedia. They present a method for document filtering for long-tail entities, which is based on using aspect-features to identify relevant documents.

Nanni et al. [91] define each section of the Wikipedia page of the entity as an aspect following [8, 43, 114]. In their work, they present a learning-to-rank based method which uses both lexical and semantic features derived from various contexts such as the sentence, paragraph and section where the entity is mentioned in text. They use two types of feature-vectors: (a) Word Vector Models, which consider the symbolic representation of each word as a token using TF-IDF and BM25, and rank aspects using the header, content and entity representations and, (b) Distributional Semantic Models, where each word/entity is represented by its embedding for ranking aspects with header and content representations. They show that using lexical and semantic features with different context sizes improves performance over several established baselines. They also showed the usefulness of their method on three downstream applications.

Recently, Ramsdell et al. [111] released a large-scale test collection for entity aspect linking along with strong baselines and example feature sets by harvesting the top-level sections from Wikipedia. Hayashi et al. [56] released “WikiAsp”, a large-scale dataset for multi-domain aspect- based summarization. WikiAsp is built using Wikipedia articles from 20 different domains, using the section titles and boundaries of each article as a proxy for aspect annotation.

3.4.2 Fine-Grained Entity Typing

A task related to entity aspect linking is fine-grained entity typing (FET): Given some text, and the span of an entity mention in this text, assign fine-grained type labels to the mention [74]. Approaches to FET are often learning-based using features that are either handcrafted [51,74] or learnt using neural networks [125,143,153]. Alternatively, entity linking is also leveraged for FET [24,58]. For example, [58] use entity linking for clustering and type name selection whereas [24] use entity linking for fine-grained entity type classification. Entity *aspects* are different from *types*: While aspects resolve the topics in which an entity is referenced (e.g., “oysters”as food versus ecosystems), types resolve which of many roles the entity can take on (e.g., food ingredient or dish).

3.4.3 Text Similarity

The entity aspect linking task may be addressed by learning the similarity between the texts of the aspect and the context. Often, text similarity is learnt by using BERT [31] to learn embeddings of the two texts such that the cosine distance of the embeddings is minimized. The similarity may also be learnt using a fully-connected layer trained jointly with the model [80]. Alternatively, pre-trained text embedding methods such as Word2Vec [87] or GloVe [102] may be used to create embeddings of the two texts for use with cosine similarity.

3.4.4 Document Retrieval

One may consider the context as a query and the aspect as a document to be retrieved. Hence, a related task is document retrieval. Some traditional, term-based document retrieval models are BM25 [60], Language Models [107], and TF-IDF [122,128]. Recently, much work has been done in the area of Neural-IR [25,48,89,98,124,148]. As the list is long, the interested reader is referred to the excellent treatise on Neural-IR by Mitra et al. [88] and Lin et al. [71].

3.4.5 Entity Relatedness

Entity relatedness measures the degree to which two entities are similar. Many entity relatedness measures have been developed [108, 120, 141, 158], based on proximity of entities in the knowledge graph, the number of in-links and out-links, etc. Entity relatedness may also be measured using the cosine similarity of the entity embeddings obtained using a graph embedding method [14, 106, 115, 155]. The entity aspect linking task can be addressed as a semantic similarity task based on the relatedness of the entities in the context and aspect.

3.4.6 Entity Linking

Entity linking systems [41,83,104,133] aim to identify and disambiguate entity mentions in text by examining the context around the entity. In this work, we further enrich an entity link with the correct aspect of the entity mentioned in the text. Several entity linking systems exist such as TagMe [41], DBpedia Spotlight [83], WAT [104], and REL [133]. All these systems examine the context around the entity to disambiguate the entity mention. In this work, we aim to further enrich an entity link with the correct aspect of the entity mentioned in the text.

3.5 Ad-Hoc Document Retrieval Using Entities

Since we utilize entity information for text retrieval, our problem is also related to the problem of ad-hoc document retrieval where semantic information in the form of entities is utilized for text retrieval. In this section, we review some methods available in the literature for leveraging entities for the document retrieval task. The approaches in literature can be grouped into three broad families as follows: Expansion-based, Projection-based and Entity-based [4]. This particular order corresponds to the temporal evolution of research in this area, where the tendency toward more and more explicit entity semantics is clearly reflected. A component common to all approaches described in this section is finding semantically related entities to a query. Three approaches are mainly used for this purpose: (1) Entities mentioned in the query, (2) Entities retrieved from a knowledge base, and (3) Entities from documents in an initial candidate set.

3.5.1 Expansion-Based Methods

These methods utilize entities as a source of expansion terms to enrich the representation of the query. In query expansion, we retrieve an initial candidate set of documents for the query and assume the top-k of this ranking to be relevant for the query. We then expand the query using terms from these top-k documents and retrieve documents using this expanded query. Akin to query expansion with terms, the idea of entity-centric query expansion is to estimate the expanded query model θ_q by using the set of query entities E_q. Meij et al. [82] propose a query expansion method based on double translation: first, translating the query to a set of relevant entities, then considering the vocabulary of terms associated with those entities as possible expansion terms to estimate the expanded query model. Xiong et al. [145] use the entity description from a knowledge base (Freebase) for the purpose of query expansion and rank documents using the expanded query.

Another approach is to use an entity language model which captures the language usage associated with the entity and represents it as a multinomial probability distribution over the vocabulary of terms. Xu et al. [154] take a linear combination of term scores across multiple entity fields. Meij et al. [82] suggest to sample the terms from documents mentioning the entity if descriptions are not available in the knowledge repository. Dalton et al. [26] propose the Entity Context Model (ECM) where a small context around the entity (such as a sentence mentioning the entity or a small window around the entity mention) is considered and all such contexts aggregated and weighted by the document retrieval score to derive a distribution over the words.

Usage of surface forms for the query entities as expansion terms is another common expansion technique [26, 75].

3.5.2 Projection-Based Methods

The vocabulary mismatch problem between queries and documents often leads to many relevant document not being retrieved by the IR system. Although query expansion can minimize this to a certain extent by bringing the original query closer to the actual information need, the problem still remains. One approach to solving the problem might be to construct a high-dimensional latent entity space and project the query and document to this entity space. The similarity between the query and document is then calculated in this space. This approach allows to uncover hidden (latent) semantic relationships between queries and documents. For example, Gabrilovitch et al. [46] propose Explicit Semantic Analysis (ESA), where each term t is represented semantically as a concept vector of length $\mid E \mid$. This vector consists of entities from a knowledge repository and the strength of the association between the term t and the given entity is given by the values in this vector. Each such value is computed by taking the TF-IDF weight of t in the description of e (in ESA, the Wikipedia article of e). A given text (bag-of-words) is represented by the centroid of the individual terms' concept vector, after normalizing these vectors to account for the differences in their lengths. Both the query and document are mapped to this ESA concept space and the similarity is found by taking the cosine similarity of their respective concept vectors. Although work on ESA has primarily focused on using Wikipedia as the underlying knowledge repository [38, 39, 45, 46], one could use any knowledge repository where there is sufficient coverage of concepts and concepts are associated with textual descriptions.

Liu et al. [76] propose Latent Entity Space (LES) which maps both queries and documents to a high-dimensional latent entity space, in which each dimension corresponds to one entity, and the relevance between the query and document is estimated based on their projections to each dimension in the latent space.

Xiong et al. [144] propose EsdRank which incorporates evidence from an external source by using terms and entities found in knowledge graphs such as Freebase or WordNet. A new ranking model called Latent-ListMLE (based on the learning to rank model called ListMLE) is used to rank documents with these objects and evidence.

3.5.3 Entity-Based Methods

These methods consider the entities in the documents explicitly and not in a latent space, together with traditional term-based representations, in the retrieval model. For example, Raviv et al. [113] propose some Entity-based Language Models (ELM) which not only use information about terms in the query and document, but also the entities. These language models are estimated using the query and the documents in the corpus. These models account simultaneously for (i) the uncertainty in entity linking — specifically, the confidence levels of entity markups; and, (ii) the balance between using term-based and entity-based information.

Similarly, Ensan et al. [40] present a Semantic Enabled Language Model (SELM). SELM addresses the task of document retrieval based on the degree of document relatedness to the meaning of a query. It is based on using an entity linking system to extract concepts (entities) from documents and queries. The document is represented as a graph where the nodes are the concepts and the edges are the relatedness relationship between two concepts. The documents are ranked by finding the conditional probability of generating the concepts observed in the query given all the document concepts and the relatedness relationships between them.

In the ELM, the words and entities are mixed together. In contrast, in the *Bag-of-Entities* representation, term-based and entity-based representations are kept apart and are used in “duet". The *Bag-of-Entities* model was proposed independently and simultaneously by Hasibi et al. [54] (for entity retrieval) and Xiong et al. [146] (for document retrieval). A line of work by Xiong et al. [146, 147, 152] is based on this bag-of-entities model. The basic idea is to construct a Bag-of-Entities vector for the query and documents using the entity annotations, and then re-rank an initial candidate set of documents for the query [146]. Two ranking models are used for this purpose: the first model ranks a document by the number of query entities it contains, and the second ranks a document by the frequency of query entities in it.

Later, two advanced models were proposed: (1) Explicit Semantic Ranking (ESR) Model [152], and (2) Word-Entity Duet (WED) Model [147]. In ESR, the relationship information from a knowledge graph is used to enable “soft matching" in the entity space. In WED, the query and documents are represented using four types of vectors: two bag-of-words vectors and two bag-of-entities vectors for the query and document respectively. Each element in these vectors corresponds to the frequency of a given term/entity in the query/document. This gives rise to four types of interactions between the query and documents: query terms to document terms, query terms to document entities, query entities to document terms and query entities to document entities. These four-way matching scores are combined using learning-to-rank.

CHAPTER 4

BACKGROUND: ATTENTION AND TRANSFORMER MODELS

4.1 Introduction

Sequence-to-sequence [129] models are deep learning models that have achieved a lot of success in tasks like machine translation, text summarization, and image captioning. Google Translate started using such a model in production in late 2016. A sequence-to-sequence model is a model that takes a sequence of items (words, letters, features of an images. . . etc) and outputs another sequence of items. In neural machine translation, a sequence is a series of words, processed one after another. The output is, likewise, a series of words.

Consider a very simple problem of predicting whether a movie review is positive or negative. Here, our input is a sequence of words, and the output is a single number between 0 and 1. If we used traditional deep neural networks, then we would typically have to encode our input text into a vector of fixed length using techniques like Bag-Of-Words, Word2Vec [87], etc. But note that here, the sequence of words is not preserved, and hence when we feed our input vector into the model, it has no idea about the order of words and thus it is missing a very important piece of information about the input. Thus, to solve this issue, Recurrent Neural Networks (RNN) came into the picture.

In essence, for any input $X = (x_0, x_1, ..., x_t)$ with a variable number of features, at each time-step, an RNN cell takes an item/token x_t as input and produces an output h_t

This chapter is mostly a compilation of information found in various excellent and popular sources on the internet. The necessary references to the original articles have been included. The interested readers are encouraged to refer to these original articles for further information. The author of this dissertation does not claim the text or figures of this chapter as his own. All credit goes to the authors of the original articles.

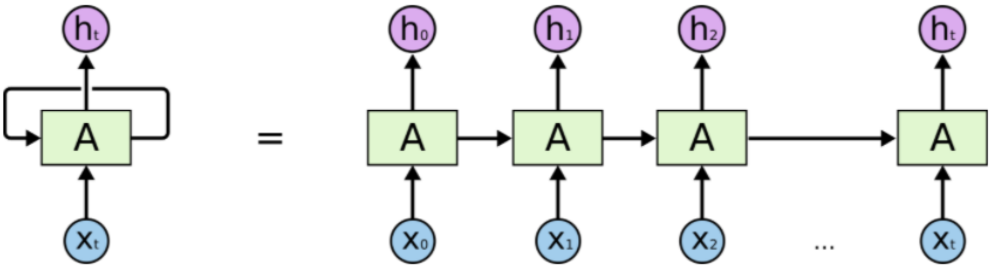


Figure 4.1: An unrolled recurrent neural network. Source: Christopher Olah [97]

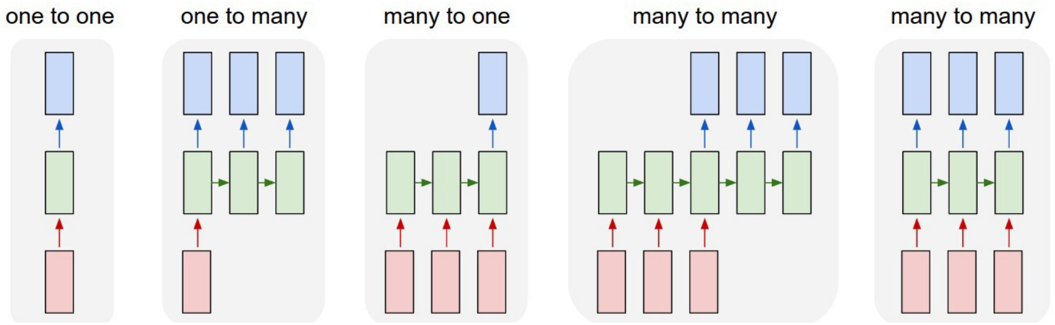


Figure 4.2: Each rectangle is a vector and arrows represent functions (e.g. matrix multiply). Input vectors are in red, output vectors are in blue and green vectors hold the RNN's state. From left to right: (1) Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification). (2) Sequence output (e.g. image captioning takes an image and outputs a sentence of words). (3) Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment). (4) Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French). (5) Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like. Source: Andrej Karpathy [64]

while passing some information onto the next time-step (see Figure 4.1). These outputs can be used according to the problem at hand. The movie review prediction problem is an example of a very basic sequence problem called many-to-one prediction. There are different types of sequence problems for which modified versions of this RNN architecture are used (see Figure 4.2). Sequence-to-Sequence (Seq2Seq) problems is a special class of Sequence Modelling Problems in which both, the input and the output, is a sequence. Encoder-Decoder models were originally built to solve such Seq2Seq problems.

In this chapter, we will be using a many-to-many type problem of Neural Machine Translation (NMT) as a running example: Given a sentence in one language (e.g., English),

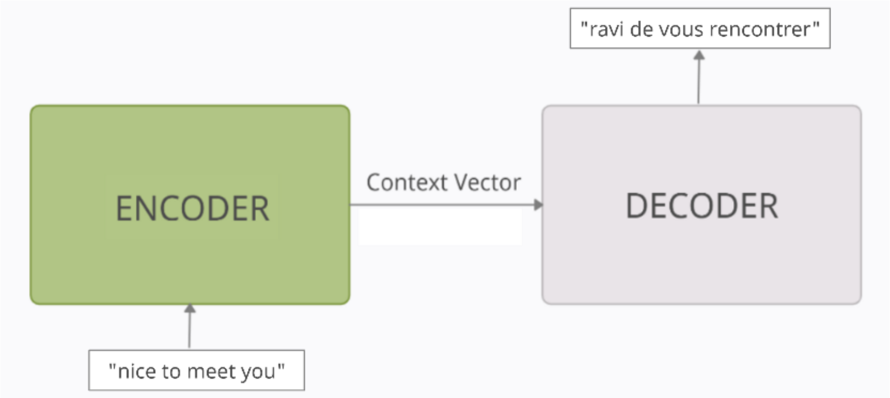


Figure 4.3: High-level view of an encoder-decoder model. The encoder takes in a sentence in English, processes it one word at a time, compiles the English sentence into a vector (called context), and passes the context vector over to the decoder. The decoder then uses this context vector to generate the French sentence, one word at a time. Source: Kriz Modes [90]

predict the the translation of the sentence to another language (e.g., Frence).

4.2 Encoder-Decoder Models

A seq2seq model consists of two components: an *encoder* and a *decoder*. The encoder processes each item in the input sequence and compiles the information it captures into a fixed-length vector (called the *context vector*). This representation is expected to be a good summary of the meaning of the *whole* source sequence. After processing the entire input sequence, the encoder sends the context vector over to the decoder, which begins producing the output sequence item by item. The encoder and decoder tend to both be recurrent neural networks (RNNs). You can set the size of the context vector when you set up your model. It is basically the number of hidden units in the encoder RNN.

An RNN cell is depicted in Figure 4.4. RNNs, by design, take two inputs at each time step: the current example they see, and a representation of the previous input (hidden state). Thus, the output at time step t depends on the current input as well as the input at time $t-1$. This is the reason they perform better when posed with sequence related tasks. The sequential information is preserved in a hidden state of the network and used in the next instance. The encoder processes the input sequence one word at a time. The word, however, needs to be represented by a vector. To transform a word into a vector, we

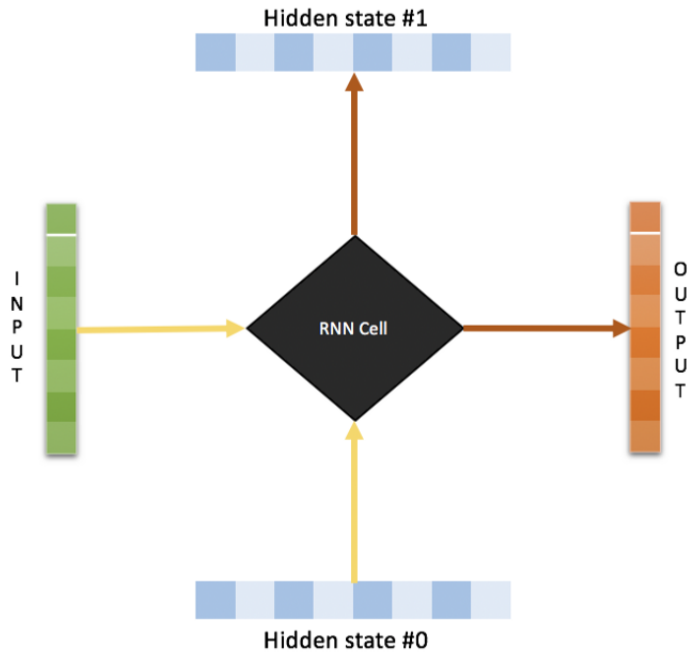


Figure 4.4: An RNN cell. RNNs take two inputs: the current example they see, and a representation of the previous input. Thus, the output at time step t depends on the current input as well as the input at time $t-1$. Source: Pranay Dugar [36]

turn to the class of methods called "word embedding" algorithms (e.g., Word2Vec). These turn words into vector spaces that capture a lot of the meaning/semantic information of the words (e.g. king - man + woman = queen). The last hidden state from the encoder is actually the context we pass along to the decoder. The decoder uses this context vector to predict a word in the sequence, and after every successive prediction, it uses the previous hidden state to predict the next word of the sequence.

4.3 Attention

The context vector turned out to be a bottleneck for these types of models. It made it challenging for the models to deal with long sentences: Often it has forgotten the first part once it completes processing the whole input. A solution was proposed in Bahdanau et al. [3] and Luong et al. [78]. These papers introduced and refined a technique called "Attention", which highly improved the quality of machine translation systems. Attention allows the model to focus on the relevant parts of the input sequence as needed. Attention is, to some extent, motivated by how we pay visual attention to different regions of an image

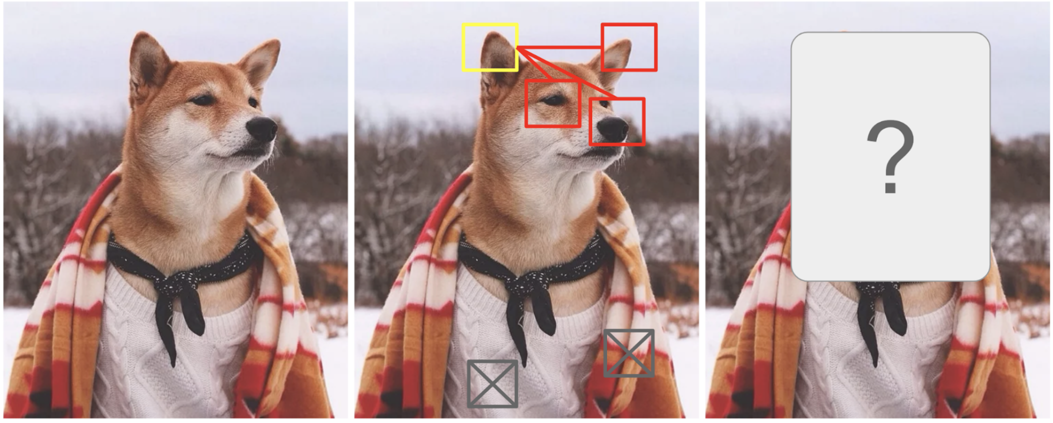


Figure 4.5: A Shiba Inu in a men's outfit. Credit: Instagram `https://www.instagram.com/mensweardog/?hl=en`

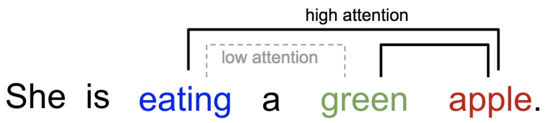


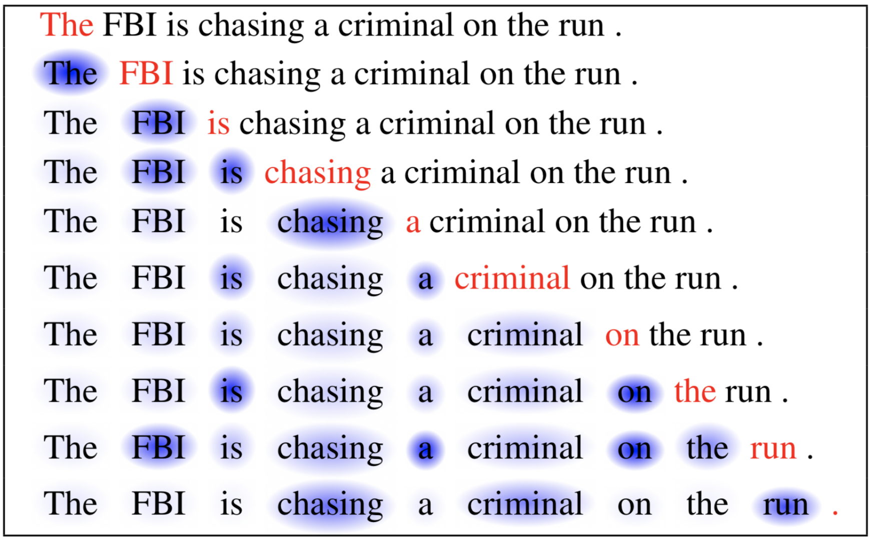
Figure 4.6: One word "attends" to other words in the same sentence differently. Source: Lilian Weng [140].

or correlate words in one sentence.

Take the picture of a Shiba Inu in Figure 4.5 as an example. Human visual attention allows us to focus on a certain region with "high resolution" (i.e. look at the pointy ear in the yellow box) while perceiving the surrounding image in "low resolution" (i.e. now how about the snowy background and the outfit?), and then adjust the focal point or do the inference accordingly. Given a small patch of an image, pixels in the rest provide clues what should be displayed there. We expect to see a pointy ear in the yellow box because we have seen a dog's nose, another pointy ear on the right, and Shiba's mystery eyes (stuff in the red boxes). However, the sweater and blanket at the bottom would not be as helpful as those doggy features.

Similarly, we can explain the relationship between words in one sentence or close context. When we see "eating" (Figure 4.6), we expect to encounter a food word very soon. The color term describes the food, but probably not so much with "eating" directly.

In a nutshell, attention in deep learning can be broadly interpreted as a vector of importance weights: in order to predict or infer one element, such as a pixel in an image or a



The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

Figure 4.7: The current word is in red and the size of the blue shade indicates the activation level. Source: Cheng et al., 2016 [21].

word in a sentence, we estimate using the attention vector how strongly it is correlated with (or "attends to" as you may have read in many papers) other elements and take the sum of their values weighted by the attention vector as the approximation of the target.

The attention mechanism was born to help memorize long source sentences in neural machine translation (NMT). Rather than building a single context vector out of the encoder's last hidden state, the secret sauce invented by attention is to create shortcuts between the context vector and the entire source input. The weights of these shortcut connections are customizable for each output element. While the context vector has access to the entire input sequence, we don't need to worry about forgetting. The alignment between the source and target is learned and controlled by the context vector.

Self-Attention. Self-attention, also known as intra-attention, is an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence. It has been shown to be very useful in machine reading, abstractive summarization, or image description generation. The Long Short-Term Memory [21] network uses self-attention to do machine reading. In Figure 4.7, the self-attention mechanism enables us to learn the correlation between the current words and the previous part of the sentence. We discuss this in more detail in Section 4.4.1.

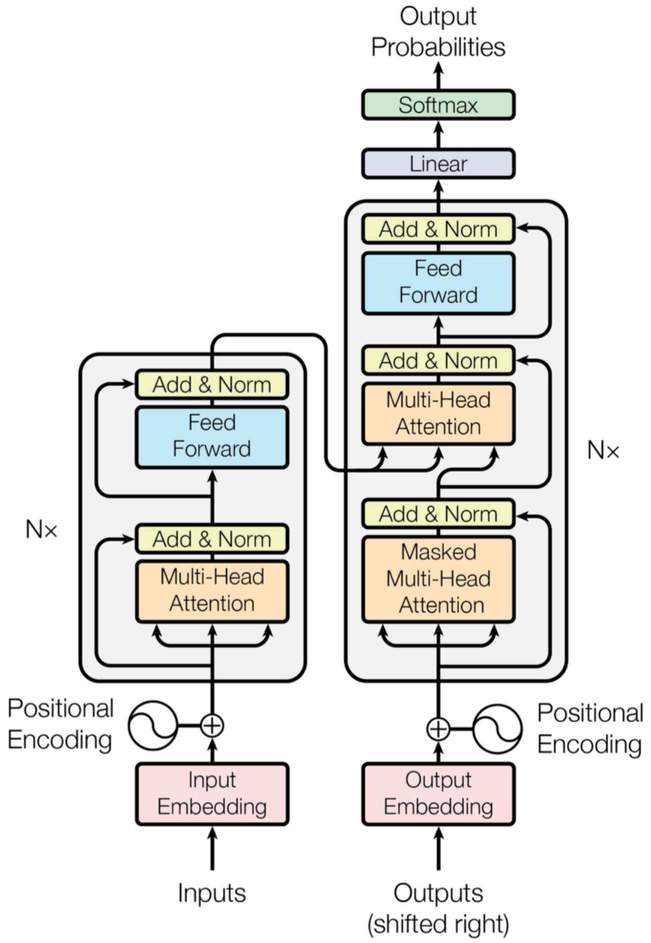


Figure 4.8: The Transformer model. Source: Vasvani et al., 2017 [134]

4.4 Transformer Model

Attention is a concept that helped improve the performance of neural machine translation applications. In this section, we will look at The Transformer – a model that uses attention to boost the speed with which these models can be trained. The Transformer outperforms the Google Neural Machine Translation model in specific tasks. The biggest benefit, however, comes from how The Transformer lends itself to parallelization. It is in fact Google Cloud's recommendation to use The Transformer as a reference model to use their Cloud TPU offering.

The Transformer was proposed by Vasvani et al. [134]. Like LSTM, Transformer is an architecture for transforming one sequence into another with the help of two parts (Encoder and Decoder), but it differs from the previously described/existing sequence-to-sequence

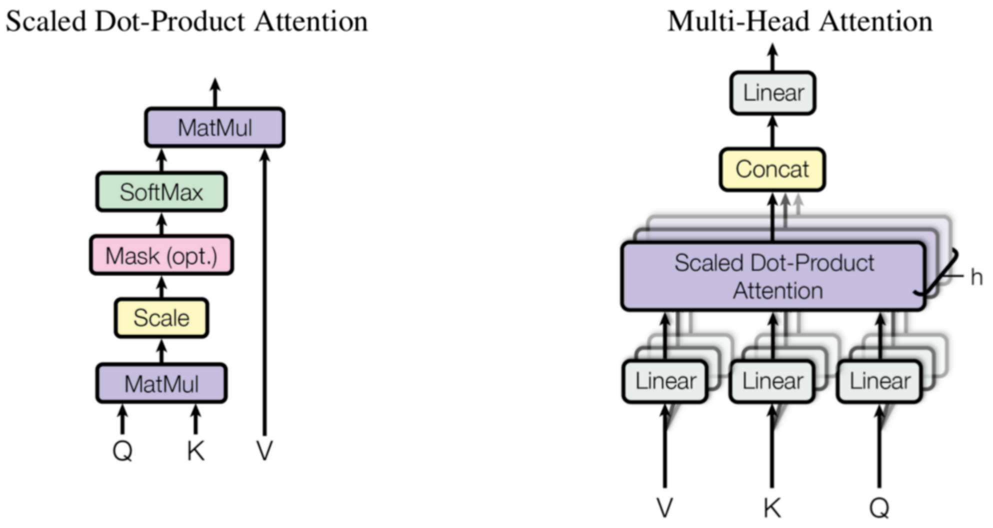


Figure 4.9: Left: Scaled Dot-Product Attention. Right: Multi-Head Attention consists of several attention layers running in parallel. Source: Vasvani et al., 2017 [134]

models because it does not use any RNNs (GRU, LSTM, etc.). RNNs were one of the best ways to capture the time dependencies in sequences. However, the paper proved that an architecture with only attention-mechanisms without any RNN can improve on the results in translation and other tasks! One improvement on natural language tasks is presented by BERT [31].

Figure 4.8 shows the Transformer model. The encoder is on the left and the decoder is on the right. Both the encoder and decoder are composed of modules that can be stacked on top of each other multiple times, which is described by $N\times$ in the figure. We see that the modules consist mainly of Multi-Head Attention and Feed Forward layers. The inputs and outputs (target sentences) are first embedded into an n-dimensional vector space since we cannot use strings directly.

One slight but important part of the model is the positional encoding of the different words. Since we have no recurrent networks that can remember how sequences are fed into a model, we need to somehow give every word/part in our sequence a relative position since a sequence depends on the order of its elements. These positions are added to the embedded representation (n-dimensional vector) of each word.

Below, we first discuss the encoder of the Transformer model in Section 4.4.1, then discuss the decoder in Section 4.4.2.

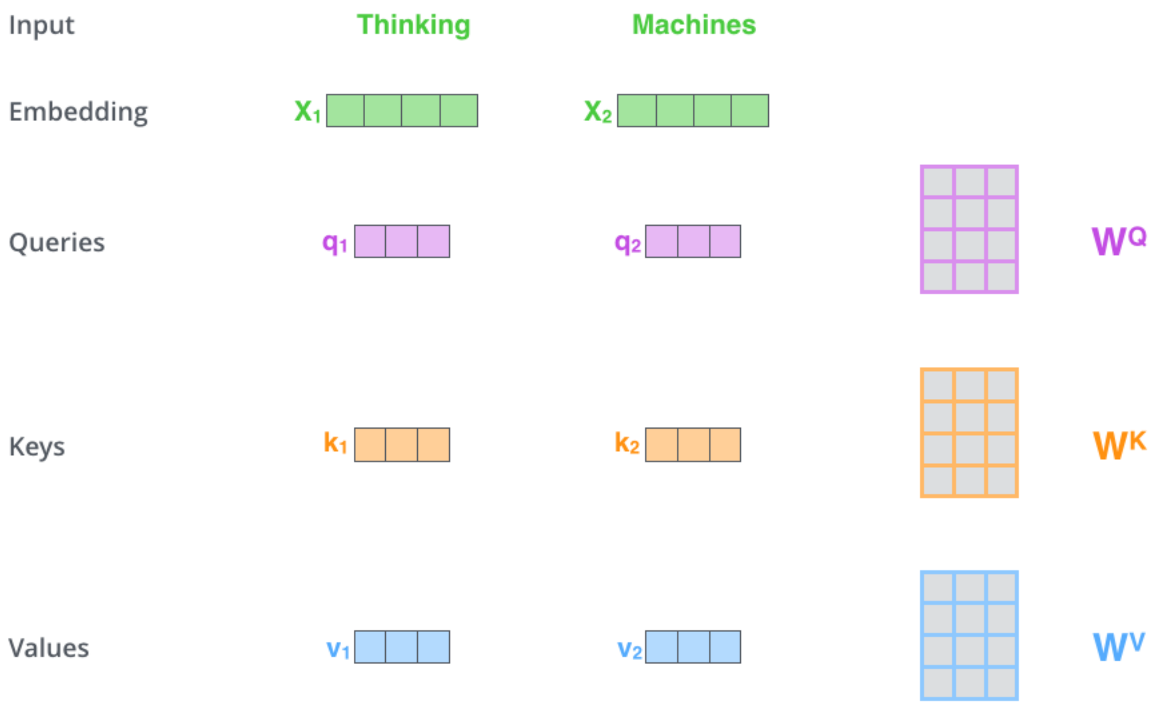


Figure 4.10: Multiplying x_1 by the W^Q weight matrix produces q_1, the "query" vector associated with that word. We end up creating a "query", a "key", and a "value" projection of each word in the input sentence. Source: Jay Alammar [2]

4.4.1 Transformer Encoder

Self-Attention

Figure 4.9 (Left) depicts the scaled-dot product self-attention used by Vasvani et al.

The **first step** in calculating self-attention is to create three vectors from each of the encoder's input vectors (in this case, the embedding of each word). So for each word, we create a Query vector, a Key vector, and a Value vector. These vectors are created by multiplying the embedding by three matrices that we trained during the training process. Notice that these new vectors are smaller in dimension than the embedding vector. Their dimensionality is 64, while the embedding and encoder input/output vectors have dimensionality of 512. They don't*have* to be smaller, this is an architecture choice to make the computation of multi-headed attention (mostly) constant. This is depicted in Figure 4.10. The “query”, “key”, and “value” vectors are abstractions that are useful for calculating and thinking about attention.

The **second step** in calculating self-attention is to calculate a score. Say we're calculat-

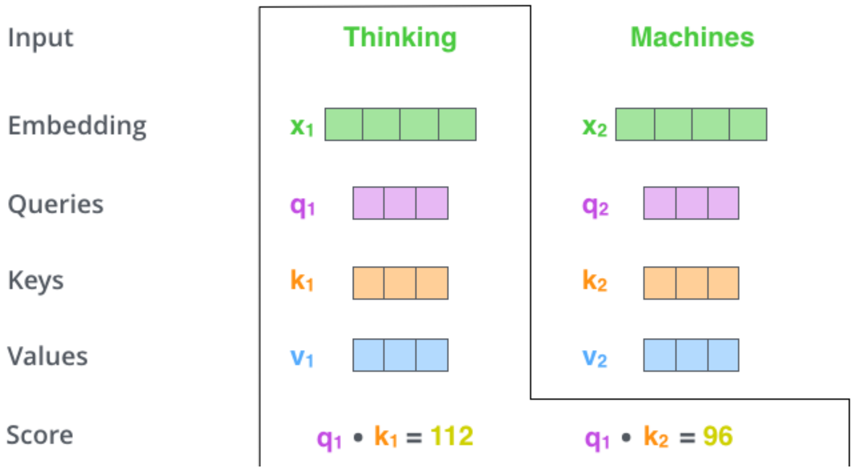


Figure 4.11: Self-attention score calculation. Source: Jay Alammar [2]

ing the self-attention for the first word in this example, “Thinking”. We need to score each word of the input sentence against this word. The score determines how much focus to place on other parts of the input sentence as we encode a word at a certain position. The score is calculated by taking the dot product of the query vector with the key vector of the respective word we are scoring. So if we are processing the self-attention for the word in position-1, the first score would be the dot product of q_1 and k_1. The second score would be the dot product of q_1 and k_2. This is depicted in Figure 4.11.

The **third and fourth steps** are to divide the scores by 8 (the square root of the dimension of the key vectors used in the paper – 64. This leads to having more stable gradients. There could be other possible values here, but this is the default), then pass the result through a softmax operation. Softmax normalizes the scores so they’re all positive and add up to 1. This softmax score determines how much each word will be expressed at this position. Clearly the word at this position will have the highest softmax score, but sometimes it’s useful to attend to another word that is relevant to the current word. This is depicted in Figure 4.12.

The **fifth step** is to multiply each value vector by the softmax score (in preparation to sum them up). The intuition here is to keep intact the values of the word(s) we want to focus on, and drown-out irrelevant words (by multiplying them by tiny numbers like 0.001, for example).

The **sixth step** is to sum up the weighted value vectors. This produces the output of

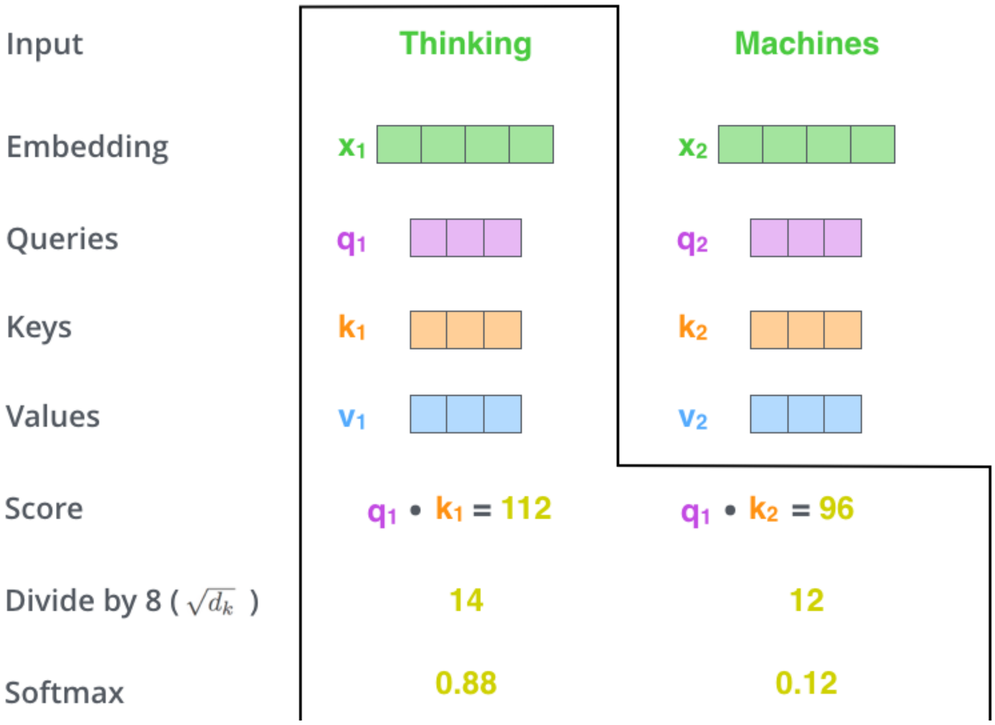


Figure 4.12: Softmax step for calculating self-attention. Source: Jay Alammar [2]

the self-attention layer at this position (for the first word). Steps 5 and 6 are depicted in Figure 4.13.

Matrix calculation of self-attention. The first step is to calculate the Query Q, Key K, and Value V matrices. We do that by packing our embeddings into a matrix X, and multiplying it by the weight matrices we have trained (W^Q, W^K, W^V). This is depicted in Figure 4.14. Finally, since we're dealing with matrices, we can condense steps two through six in one formula to calculate the outputs of the self-attention layer:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V$$

Multi-Head Self-Attention

The paper further refined the self-attention layer by adding a mechanism called "multi-headed" attention. This improves the performance of the attention layer in two ways:

1. It expands the model's ability to focus on different positions. Yes, in the example above, z_1 contains a little bit of every other encoding, but it could be dominated by

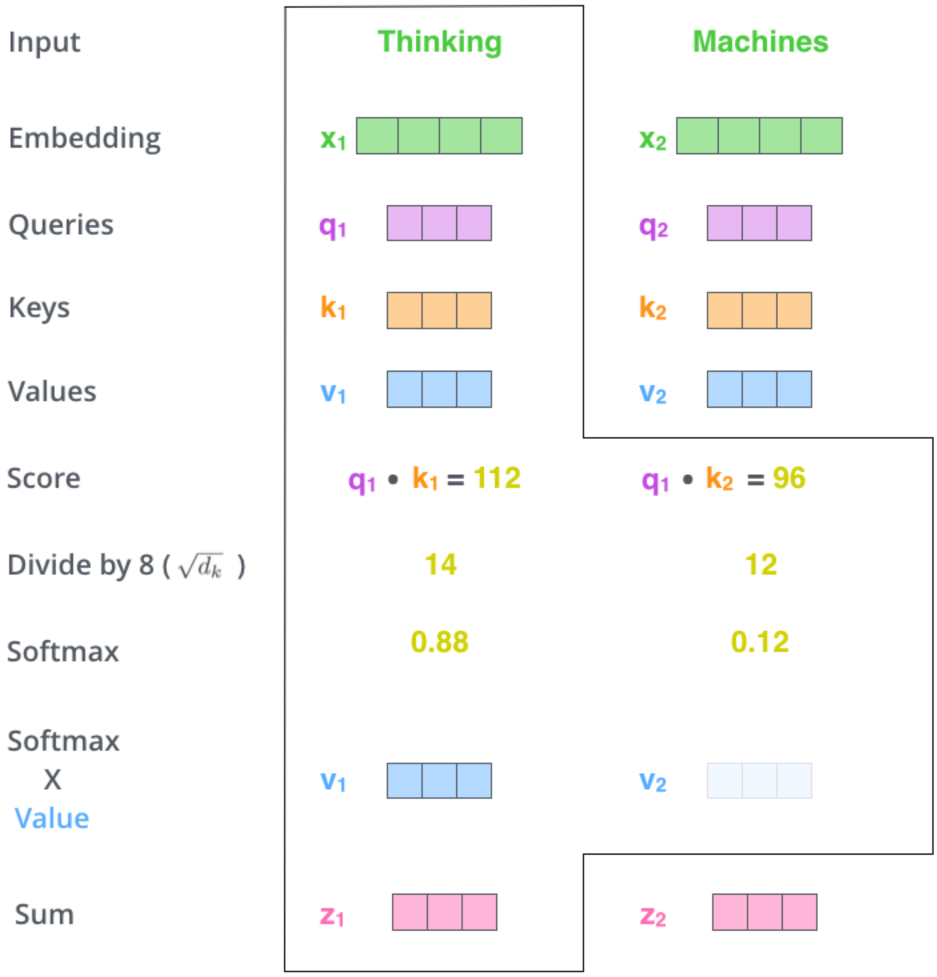


Figure 4.13: Final score for self-attention. Source: Jay Alammar [2]

the the actual word itself. It would be useful if we are translating a sentence like "The animal did not cross the street because it was too tired", we would want to know which word "it" refers to.

2. It gives the attention layer multiple "representation subspaces". As we will see next, with multi-headed attention we have not only one, but multiple sets of Query/Key/-Value weight matrices. The Transformer uses eight attention heads, so we end up with eight sets for each encoder/decoder (Figure 4.15). Each of these sets is randomly initialized. Then, after training, each set is used to project the input embeddings (or vectors from lower encoders/decoders) into a different representation subspace.

If we do the same self-attention calculation we outlined above, just eight different times

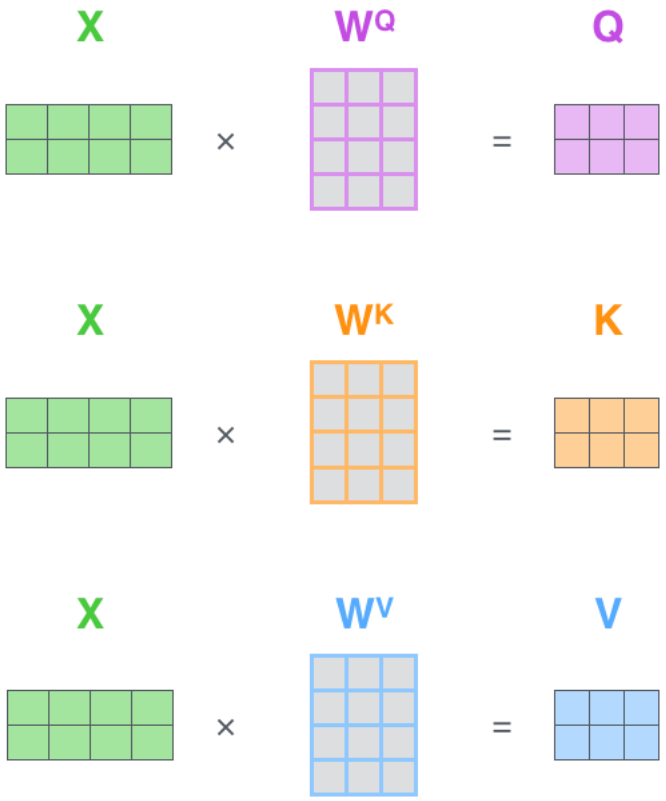


Figure 4.14: Calulating self-attention using matrices. Every row in the matrix X corresponds to a word in the input sentence. Source: Jay Alammar [2]

with different weight matrices, we end up with eight different Z matrices (Figure 4.16). This leaves us with a bit of a challenge: The feed-forward layer is not expecting eight matrices – it's expecting a single matrix (a vector for each word). So we need a way to condense these eight down into a single matrix. How do we do that? We concatinate the matrices, then multiply them by an additional weights matrix W^O (Figure 4.17). That's pretty much all there is to multi-headed self-attention. The process is put together in Figure 4.18.

Positional Encoding

One thing that's missing from the model as we have described it so far is a way to account for the order of the words in the input sequence. To address this, the transformer adds a vector to each input embedding (Figure 4.19). These vectors follow a specific pattern that the model learns, which helps it determine the position of each word, or the distance between different words in the sequence. The intuition here is that adding these values to

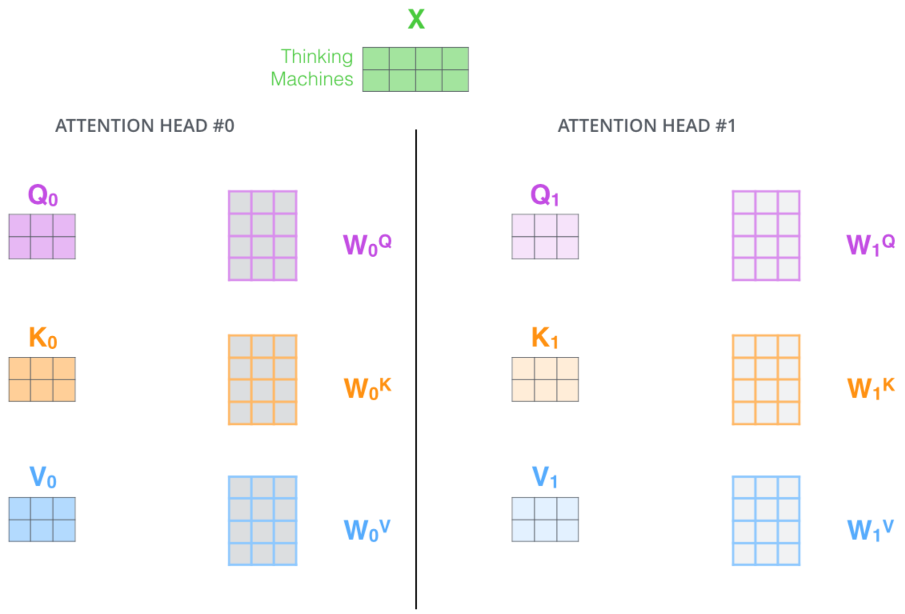


Figure 4.15: With multi-headed attention, we maintain separate $Q/K/V$ weight matrices for each head resulting in different $Q/K/V$ matrices. As we did before, we multiply X by the $W^Q/W^K/W^V$ matrices to produce $Q/K/V$ matrices. Source: Jay Alammar [2]

the embeddings provides meaningful distances between the embedding vectors once they are projected into $Q/K/V$ vectors and during dot-product attention.

4.4.2 Transformer Decoder

Now that we have covered most of the concepts on the encoder side, we basically know how the components of decoders work as well. But let's take a look at how they work together. The encoder starts by processing the input sequence. The output of the top encoder is then transformed into a set of attention vectors K and V. These are to be used by each decoder in its "encoder-decoder attention" layer which helps the decoder focus on appropriate places in the input sequence.

The following steps repeat the process until a special symbol is reached indicating the transformer decoder has completed its output. The output of each step is fed to the bottom decoder in the next time step, and the decoders bubble up their decoding results just like the encoders did. And just like we did with the encoder inputs, we embed and add positional encoding to those decoder inputs to indicate the position of each word.

The self attention layers in the decoder operate in a slightly different way than the one

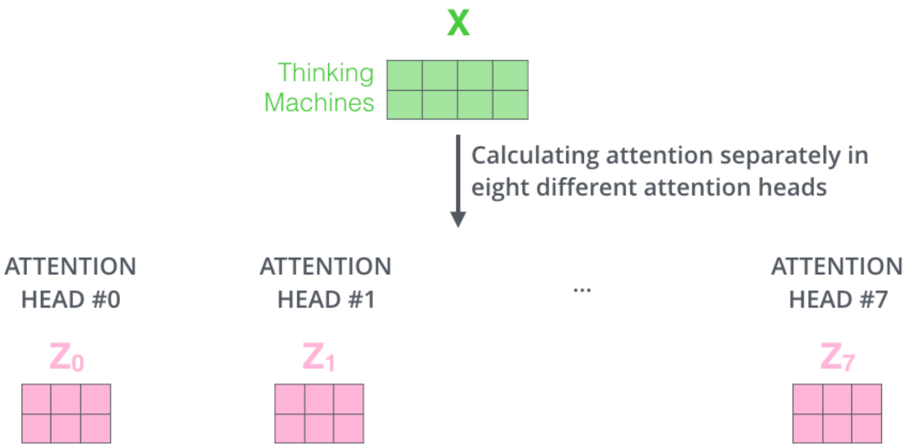


Figure 4.16: Calculating the attention separately in eight different attention heads. Source: Jay Alammar [2]

in the encoder: In the decoder, the self-attention layer is only allowed to attend to earlier positions in the output sequence. This is done by masking future positions (setting them to negative infinity) before the softmax step in the self-attention calculation.

The “Encoder-Decoder Attention” layer works just like multi-headed self-attention, except it creates its Queries matrix from the layer below it, and takes the Keys and Values matrix from the output of the encoder stack.

4.4.3 The Final Linear and Softmax Layer

The decoder stack outputs a vector of floats. How do we turn that into a word? That’s the job of the final Linear layer which is followed by a Softmax Layer. The Linear layer is a simple fully connected neural network that projects the vector produced by the stack of decoders, into a much, much larger vector called a logits vector. Let’s assume that our model knows 10,000 unique English words (our model’s “output vocabulary”) that it’s learned from its training dataset. This would make the logits vector 10,000 cells wide – each cell corresponding to the score of a unique word. That is how we interpret the output of the model followed by the Linear layer. The softmax layer then turns those scores into probabilities (all positive, all add up to 1.0). The cell with the highest probability is chosen, and the word associated with it is produced as the output for this time step.

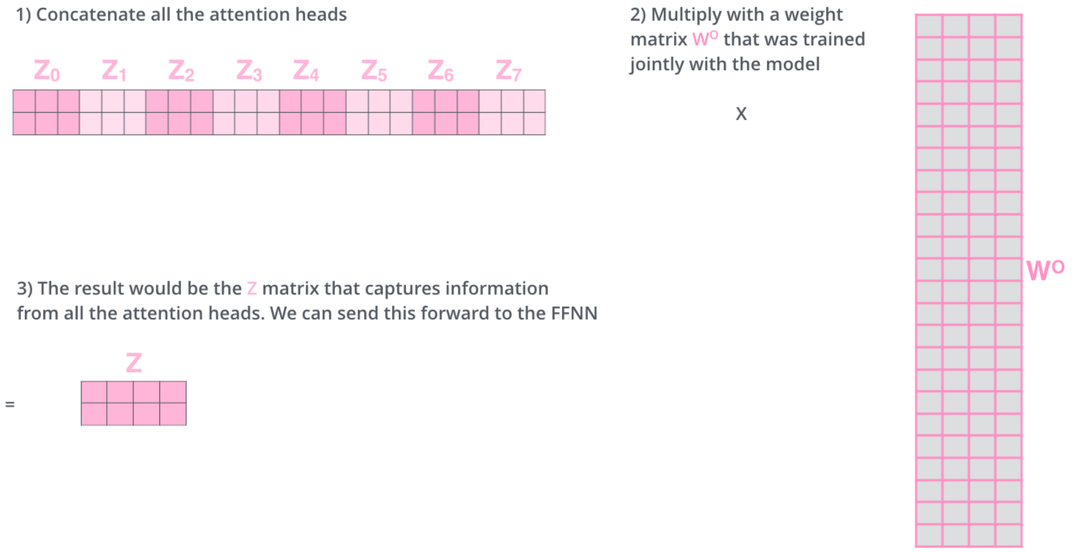

1) Concatenate all the attention heads
Z_0 Z_1 Z_2 Z_3 Z_4 Z_5 Z_6 Z_7
2) Multiply with a weight matrix W^O that was trained jointly with the model
X
W^O
3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN
Z
=

Figure 4.17: Calculating the final Z matrix. Source: Jay Alammar [2]

4.5 BERT: Bidirectional Encoder Representations from Transformers

4.5.1 Introduction

One of the biggest challenges in NLP is the lack of enough training data. Overall, there is enormous amount of text data available, but if we want to create task-specific datasets, we need to split that pile into the very many diverse fields. And when we do this, we end up with only a few thousand or a few hundred thousand human-labeled training examples. Unfortunately, in order to perform well, deep learning based NLP models require much larger amounts of data — they see major improvements when trained on millions, or billions, of annotated training examples.

In the field of computer vision, researchers have repeatedly shown the value of transfer learning–pre-training a neural network model on a known task, for instance ImageNet, and then performing fine-tuning using the trained neural network as the basis of a new purpose-specific model. In recent years, researchers have been showing that a similar technique can be useful in many natural language tasks. These general purpose pre-

Section based on articles by Samia Khalid [66] and Rani Horev [57].

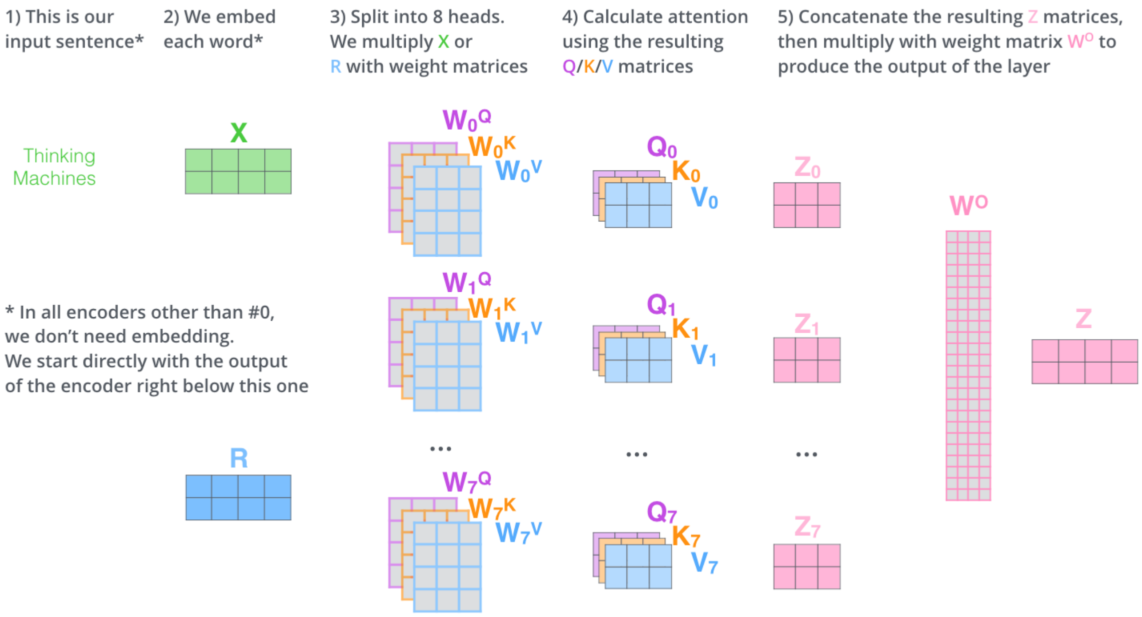

1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer
Thinking Machines
X
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
R
W_0^Q
W_0^K
W_0^V
W_1^Q
W_1^K
W_1^V
...
W_7^Q
W_7^K
W_7^V
Q_0
K_0
V_0
Q_1
K_1
V_1
...
Q_7
K_7
V_7
Z_0
Z_1
...
Z_7
W^O
Z

Figure 4.18: Multi-head self-attention end-to-end. Source: Jay Alammar [2]

trained models can then be fine-tuned on smaller task-specific datasets, e.g., when working with problems like question answering and sentiment analysis. This approach results in great accuracy improvements compared to training on the smaller task-specific datasets from scratch. BERT [31] is a recent addition to these techniques for NLP pre-training. It has caused a stir in the Machine Learning community by presenting state-of-the-art results on a wide variety of NLP tasks, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others.

4.5.2 Core Idea Behind BERT

Given a sentence such as "The woman went to the store and bought a _____ of shoes", a language model might complete this sentence by saying that the word "cart" would fill the blank 20% of the time and the word "pair" 80% of the time. In the pre-BERT world, a language model would have looked at this text sequence during training from either left-to-right or combined left-to-right and right-to-left. This one-directional approach works well for generating sentences — we can predict the next word, append that to the sequence, then predict the next to next word until we have a complete sentence.

BERT's key technical innovation is applying the bidirectional training of Transformer, a

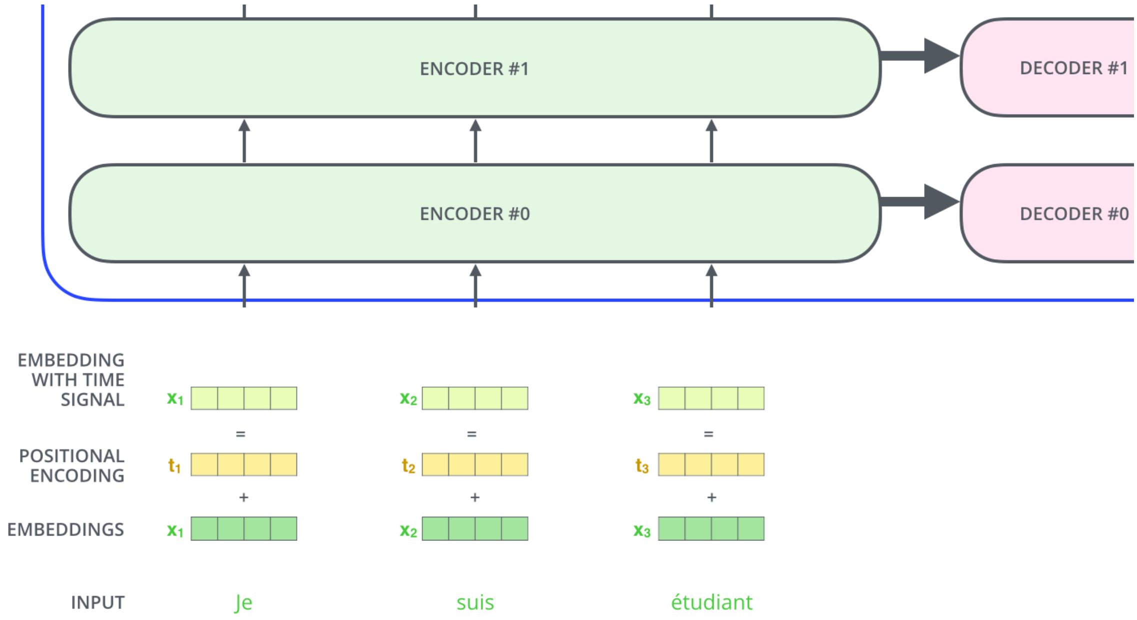


Figure 4.19: To give the model a sense of the order of the words, we add positional encoding vectors – the values of which follow a specific pattern. Source: Jay Alammar [2]

popular attention model, to language modelling. This means we can now have a deeper sense of language context and flow compared to the single-direction language models. Instead of predicting the next word in a sequence, BERT makes use of a novel technique called Masked LM (MLM): it randomly masks words in the sentence and then it tries to predict them. Masking means that the model looks in both directions and it uses the full context of the sentence, both left and right surroundings, in order to predict the masked word. Unlike the previous language models, it takes both the previous and next tokens into account at the same time. The existing combined left-to-right and right-to-left LSTM based models were missing this “same-time part”. (It might be more accurate to say that BERT is non-directional though.)

Why is this non-directional approach so powerful? Pre-trained language representations can either be *context-free* or *context-based*. Context-based representations can be unidirectional or bidirectional. Context-free models like word2vec [87] generate a single word embedding representation (a vector of numbers) for each word in the vocabulary. For example, the word “bank” would have the same context-free representation in “bank account” and “bank of the river”. On the other hand, context-based models generate a representation of each word that is based on the other words in the sentence. For example, in the sentence “I accessed the bank account”, a unidirectional contextual model would represent “bank” based on “I accessed the” but not “account”. However, BERT represents “bank” using both its previous and next context — “I accessed the _____ account” — starting from the very bottom of a deep neural network, making it deeply bidirectional.

Moreover, BERT is based on the Transformer model architecture, instead of LSTMs. As we saw in Chapter 4, a Transformer applies an attention mechanism to understand relationships between all words in a sentence, regardless of their respective position. For example, given the sentence, “I arrived at the bank after crossing the river”, to determine that the word “bank” refers to the shore of a river and not a financial institution, the Transformer can learn to immediately pay attention to the word “river” and make this decision in just one step.

4.5.3 How Does BERT Work?

BERT relies on a Transformer (the attention mechanism that learns contextual relationships between words in a text). As discussed in Chapter 4, a basic Transformer consists of an encoder to read the text input and a decoder to produce a prediction for the task. Since BERT’s goal is to generate a language representation model, it only needs the encoder part. The input to the encoder for BERT is a sequence of tokens, which are first converted into vectors and then processed in the neural network.

BERT needs the input to be decorated with some extra metadata:

1. **Token embeddings:** A [CLS] token is added to the input word tokens at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence.

2. **Segment embeddings:** A marker indicating Sentence A or Sentence B is added to each token. This allows the encoder to distinguish between sentences.

3. **Positional embeddings:** A positional embedding is added to each token to indicate its position in the sentence.

Essentially, the Transformer stacks a layer that maps sequences to sequences, so the output is also a sequence of vectors with a 1:1 correspondence between input and output tokens at the same index. And as we learnt earlier, BERT does not try to predict the next word in the sentence. Training makes use of the following two strategies:

Masked Language Modelling (MLM). The idea here is: Randomly mask out 15% of the words in the input – replacing them with a [MASK] token – run the entire sequence through the BERT attention based encoder and then predict only the masked words, based on the context provided by the other non-masked words in the sequence. However, there is a problem with this naive masking approach: the model only tries to predict when the [MASK] token is present in the input, while we want the model to try to predict the correct tokens regardless of what token is present in the input. To deal with this issue, out of the 15% of the tokens selected for masking, 80% of the tokens are actually replaced with the token [MASK], 10% tokens are replaced with a random token, and the remaining 10% tokens are left unchanged. While training, the BERT loss function considers only the prediction of the masked tokens and ignores the prediction of the non-masked ones. This results in a model that converges much more slowly than left-to-right or right-to-left models.

Next Sentence Prediction (NSP). In order to understand the relationship between two sentences, BERT's training process also uses next sentence prediction. A pre-trained model with this kind of understanding is relevant for tasks like question answering. During training, the model gets pairs of sentences as input, and it learns to predict if the second sentence is the next sentence in the original text as well.

As we have seen earlier, BERT separates sentences with a special [SEP] token. During training, the model is fed with two input sentences at a time such that 50% of the time the second sentence comes after the first one, and 50% of the time, it is a a random sentence from the full corpus. BERT is then required to predict whether the second sentence is random or not, with the assumption that the random sentence will be disconnected from the first sentence.

PART II

CONTRIBUTIONS

This page intentionally left blank.

Chapter 5

Entity-Support Passage Retrieval for Entity Retrieval

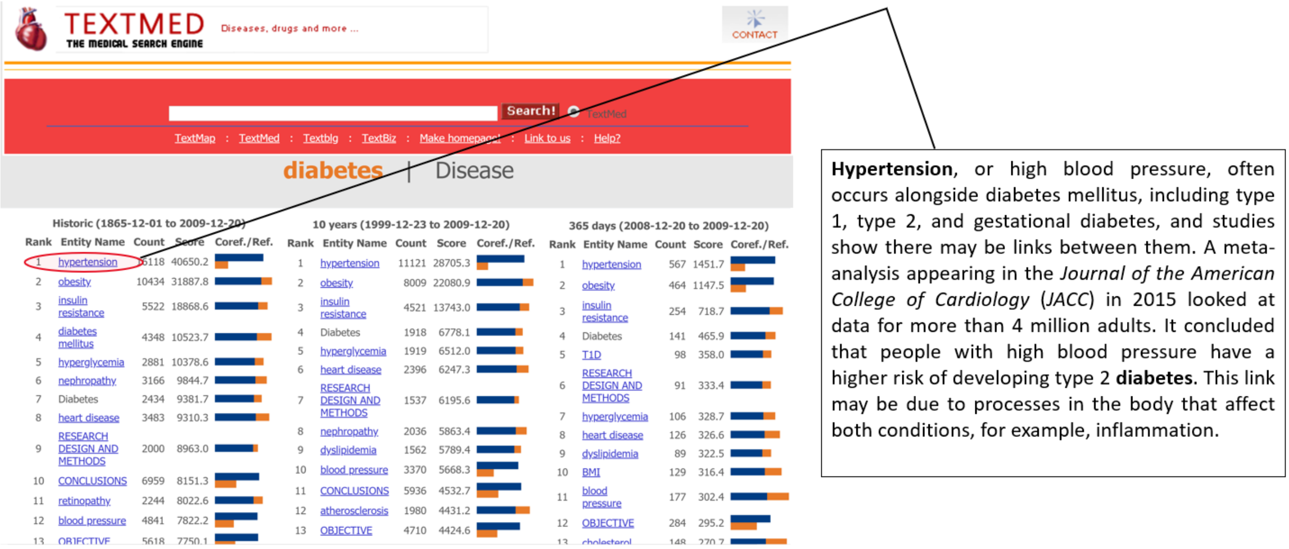
5.1 Introduction

5.1.1 Motivation

Search engines have become ubiquitous in the present world. While large-scale commercial services such as Amazon (which allows us to search for products) have integrated entity ranking algorithms into their systems, they still lack the “snippet retrieval” feature which is ubiquitous in document retrieval systems: Document retrieval systems such as Google display a snippet of text along with the “ten blue links” in response to a user’s information need to help the user decide if they are interested in the content of the document pointed to by the link. Such search snippets play an important role in guiding users to the right documents [130]. While retrieval of entities from knowledge graphs (such as Freebase and DBpedia) is well-studied, it is an open problem how to extract search snippets for knowledge graph entities, especially when the short description of the entity (often the introductory paragraph from the entity’s Wikipedia page) is not a meaningful explanation of relevance [35].

Entity ranking as a task has been extensively studied in the past [7, 12, 28, 126, 137]. Given a user’s information need (henceforth *query*) along with a Knowledge Graph, the

Part of this chapter is published as: Shubham Chatterjee and Laura Dietz. 2019. Why Does This Entity Matter? Support Passage Retrieval for Entity Retrieval. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR ’19)*. Association for Computing Machinery, New York, NY, USA, 221–224. https://doi.org/10.1145/3341981.3344243


TEXTMED
THE MEDICAL SEARCH ENGINE
Diseases, drugs and more ...
CONTACT
Search!
TextMed
TextMap : TextMed : Textblg : TextBiz : Make homepage! : Link to us : Help?
diabetes | Disease

Historic (1865-12-01 to 2009-12-20)

Rank	Entity Name	Count	Score	Coref./Ref.
1	hypertension	16118	40650.2	
2	obesity	10434	31887.8	
3	insulin resistance	5522	18868.6	
4	diabetes mellitus	4348	10523.7	
5	hyperglycemia	2881	10378.6	
6	nephropathy	3166	9844.7	
7	Diabetes	2434	9381.7	
8	heart disease	3483	9310.3	
9	RESEARCH DESIGN AND METHODS	2000	8963.0	
10	CONCLUSIONS	6959	8151.3	
11	retinopathy	2244	8022.6	
12	blood pressure	4841	7822.2	
13	OBJECTIVE	5618	7750.1	

10 years (1999-12-23 to 2009-12-20)

Rank	Entity Name	Count	Score	Coref./Ref.
1	hypertension	11121	28705.3	
2	obesity	8009	22080.9	
3	insulin resistance	4521	13743.0	
4	Diabetes	1918	6778.1	
5	hyperglycemia	1919	6512.0	
6	heart disease	2396	6247.3	
7	RESEARCH DESIGN AND METHODS	1537	6195.6	
8	nephropathy	2036	5863.4	
9	dyslipidemia	1562	5789.4	
10	blood pressure	3370	5668.3	
11	CONCLUSIONS	5936	4532.7	
12	atherosclerosis	1980	4431.2	
13	OBJECTIVE	4710	4424.6	

365 days (2008-12-20 to 2009-12-20)

Rank	Entity Name	Count	Score	Coref./Ref.
1	hypertension	567	1451.7	
2	obesity	464	1147.5	
3	insulin resistance	254	718.7	
4	Diabetes	141	465.9	
5	T1D	98	358.0	
6	RESEARCH DESIGN AND METHODS	91	333.4	
7	hyperglycemia	106	328.7	
8	heart disease	126	326.6	
9	dyslipidemia	89	322.5	
10	BMI	129	316.4	
11	blood pressure	177	302.4	
12	OBJECTIVE	284	295.2	
13	cholesterol	148	270.7	

Hypertension, or high blood pressure, often occurs alongside diabetes mellitus, including type 1, type 2, and gestational diabetes, and studies show there may be links between them. A meta-analysis appearing in the *Journal of the American College of Cardiology* (*JACC*) in 2015 looked at data for more than 4 million adults. It concluded that people with high blood pressure have a higher risk of developing type 2 **diabetes**. This link may be due to processes in the body that affect both conditions, for example, inflammation.

Figure 5.1: An example support passage for the entity "Hypertension" relevant to the query *Diabetes*. This support passage explains how the entity is related to the information need. Without this passage, the entity ranking does not make much sense to a person who does not have knowledge about Hypertension and Diabetes.

entity ranking task is to retrieve and rank entities from this Knowledge Graph in order of relevance of the entity to the query. Several applications display a ranking of entities for a user information need. For example, TextMed[1] (Figure 5.1) is a search engine that displays a ranking of entities for a medical information need such as *Diabetes*. As shown in Figure 5.1, the connection between the top-ranked entity "Hypertension" and the query *Diabetes* is not clear from this ranking. Tombros et al. [130] have shown that in document retrieval systems, presenting the users with a short textual description summarizing the document helps them judge the importance and utility of the results. Analogously, we want to present a short text passage such as the one shown on the right in Figure 5.1 that explains the connection between entity "Hypertension" and the query *Diabetes* to a user.

Entity-Support Passage Retrieval Task. Given a user's information need Q; an external system predicts a ranking of entities E. For every relevant entity $e_i \in E$, we want to retrieve and rank K passages s_{ik} which explain why this entity e_i is relevant for Q. We call the entity e_i *target entity*, and the passage s_{ik} *entity-support passage*.

[1] http://www.textmed.com/

5.1.2 Why Do We Need Entity-Support Passages?

Entity-oriented search systems often use the introductory paragraph from an entity's Wikipedia page as the entity's description [77, 80, 151]. Hence, the curious reader might wonder: Is the introductory paragraph not sufficient to serve as an entity's support passage? This question has been studied and answered in the past: Dietz et al. [35] found that in less than 50% cases, the introductory paragraph from an entity's Wikipedia page is useful for explaining the connection between the query and entity. In another experiment conducted by the organizers of the TREC Complex Answer Retrieval [34] track, the assessors for the entity retrieval task were provided with the introductory paragraph from the Wikipedia article of the corresponding entity and asked to judge whether the introductory paragraph was sufficient to explain the connection between the query and the entity. It was found that the introductory paragraph was generally not relevant to the entity in the context of the query, and hence insufficient to explain why the entity is relevant to the query.

5.1.3 Importance to the Research Community

As such, the entity-support passage retrieval task (in various flavors) has received much attention from the IR research community. In fact, the task as defined above was part of the entity retrieval task of TREC Complex Answer Retrieval track [33] where the participants were required to retrieve Wikipedia entities in response to a query, along with passages from Wikipedia which explain how/why the entity is relevant/related to the query. Similarly, the 2020 edition of the TREC News [127][2] track explored a variation of the above task for the news domain: link the entities present in the text from a news article to an external resource such as Wikipedia which provides more information on the entity. The Retrieval From Conversational Dialogues (RCD)[3] track at FIRE 2020 explored a variation of the above task for the conversational IR domain: return a ranked list of passages from Wikipedia containing information on the entities in a dialogue.

[2] http://trec-news.org/

[3] https://rcd2020firetask.github.io/RCD2020FIRETASK/

Query: Cholera
Entity: Oyster

Most cholera cases in developed countries are a result of transmission by food. Food transmission can occur when people harvest seafood such as **oysters** in waters infected with sewage, as Vibrio cholerae accumulates in planktonic crustaceans and the **oysters** eat the zooplankton.

Oyster is the common name for a number of different families of salt-water bivalve molluscs that live in marine or brackish habitats. **Oysters** influence nutrient cycling, water filtration, habitat structure, biodiversity, and food web dynamics.

Cholera is an infection of the small intestine by some strains of the bacterium Vibrio cholerae. The classic symptom is large amounts of watery diarrhea that lasts a few days. Vomiting and muscle cramps may also occur. Cholera can be caused by eating **oysters**.

Figure 5.2: Example query and entity with support passages. Left: The passage is relevant to the query and entity, and the entity is salient in the passage. The passage clarifies how oysters may cause cholera. Hence, this is a good support passage for the query-entity pair. Middle: The passage is relevant to the entity but not to the query. Right: The passage is relevant to the query but not to the entity as the entity is not salient in the passage. The passages in the middle and right are not good support passages.

5.1.4 Answering Topical Queries using Entity-Support Passages

In addition to providing more information about entities in a ranking, the entity-support passages may also be utilized in a larger end-to-end information retrieval system which aims to answer information needs of users about (yet) unfamiliar topics for which no Wikipedia articles exist (yet). To this end, one may structure the answer-space for such queries through the entities relevant to the query by assuming that the entities are the coarse-grained *subtopics* that a human would talk about while discussing about the topic. For example, while discussing about Diabetes[4], one may talk about how diabetes is treated using insulin, how it is related to heart diseases, etc. Then, the support passages for each of these entities (“insulin”, “heart disease”, etc.) may serve as text to be clustered and summarized to generate the Wikipedia-like article on the topic to be presented to the user. Hence, in this dissertation, we use information about relevant entities to retrieve relevant text for the query.

[4] This is just an illustrative example. Of course, a Wikipedia article about diabetes exists and can be found at `https://en.wikipedia.org/wiki/Diabetes`.

5.1.5 Research Gap

The current state-of-the-art for entity-support passage retrieval [13, 61] uses methods based on entity statistics such as frequency (number of candidate support passages mentioning the target entity), the KL-Divergence between the query and collection distributions, relation extraction, etc. However, for entity-support passage retrieval, it is essential to identify the relevant connections between the query and the entity. For example, in Figure 5.2, the target entity "Oyster" is relevant to the query *Cholera* because cholera may be caused in humans by consuming oysters that feed on cholera-causing bacteria called "Vibrio Cholerae": The relevant connection between the query *Cholera* and the target entity "Oyster" is through the query-relevant entity "Vibrio Cholerae". In this work, we identify such relevant connections (entities) between the query and entity and hypothesize that a text passage containing many such relevant connections (entities) would be a good support passage for the target entity.

In addition to being relevant for the query, each support passage should mention the target entity in a *salient* way. Salient means that the entity is *central* to the discussion in the passage and not just mentioned as an aside. For example, the target entity "Oyster" is salient in the left passage in Figure 5.2 but not in the right passage. The current state-of-the-art for entity-support passage retrieval does not consider the salience of the target entity in the support passage. Hence, such methods might retrieve the right passage instead of the left passage in Figure 5.2. In this work, we incorporate the salience of the target entity in a candidate support passage, and study the impact of using entity salience for the entity-support passage retrieval task.

5.1.6 Contributions

We make the following contributions through this work:

1. We propose a new model for entity-support passage retrieval called *Entity Prominence*. We show that our approach achieves new state-of-the-art results for entity-support passage retrieval by improving retrieval effectiveness by 80% (in terms of

Mean Average Precision) on average, on two publicly available datasets from TREC Complex Answer Retrieval.

2. We show that entity salience is a useful indicator and can improve retrieval effectiveness by 70% (in terms of Mean Average Precision) on average on two publicly available datasets.

3. We show that the performance on the task is dependent upon the type of context used for the entity: Performance improvements are obtained by replacing a query-independent context such as the Wikipedia article of the target entity with a query-dependent context.

5.1.7 Outline

The remainder of this chapter is organized as follows. In Sections 5.2 through 5.4, we describe our proposed method in detail. In Sections 5.5 and 5.6, we discuss the evaluation protocol and results from this work. Finally, we end the chapter with Section 5.7.

5.2 Overarching Ideas

Given a ranked list of entities for a query, we seek to embellish it with passages which would explain to the user why the entity is relevant to the query. We call the entities in the ranking as *target entities*. We only try to predict support passages for target entities which are also relevant to the query (according to an entity ground truth).

The overarching idea underlying this work is that a good support passage is (1) one where the target entity e_i is salient i.e., central to the discussion in the text, and (2) which contains many relevant connections between the query and the target entity (for example, the left passage in Figure 5.2).

Incorporating entity salience into our model is trivial: We use an off-the-shelf state-of-the-art salience detection system called SWAT [109] to incorporate entity salience into our model. Entity salience has been well-studied in the past [37, 109, 150], and several tools exist that given a text passage, identify salient entities in the passage. Our goal is not to propose a new entity salience detection system. On the contrary, we want to study whether entity salience is important for the entity-support passage retrieval task.

The important and non-trivial question is: How do we identify relevant connections between the query and the target entity? To answer this question, we note that our queries are mainly topics such as *Diabetes* or *Cholera*. As such, the connections between a topical query and a target entity may be defined in terms of other sub-topics that one might mention while discussing about the target entity in the context of the query. For example, while discussing about the target entity "Oyster" in the context of the topical query *Diabetes*, one may mention query-relevant sub-topics such as "Seafood" and "Vibrio Cholerae" related to the entity "Oyster". Our assumption is that a query-relevant text passage containing many such sub-topics would be a good support passage for the entity "Oyster".

In this work, we use the query-relevant entities e_x which *frequently co-occur with the target entity* e_i in a query-relevant context $\mathcal{C}$ as surrogates for coarse-grained sub-topics that are relevant for a discussion about the target entity e_i in the context of a topical query. We use a query-relevant candidate set of passages $\mathcal{D}$ to define an entity's query-relevant context $\mathcal{C}$, and the query-relevant entities e_x to define the relevant connections between the query and the target entity e_i. We hypothesize that a query-relevant text passage $p \in \mathcal{D}$ mentioning many such query-relevant entities (connections) e_x would be good support passage for the entity.

Our assumption about good entity-support passages implies that the support passage must be relevant to both the query and the target entity. Hence, at the heart of our approach is a model which given a query Q and a target entity e_i, combines information from two types of indicators: (1) The sub-topics pertaining to the target entity e_i which are relevant in the context of the query Q, obtained using a query-relevant context $\mathcal{C}$ of the target entity, and (2) An entity salience indicator which indicates whether the target entity e_i is salient in the support passage. These indicators are then combined in a supervised manner to score a support passage p with respect to its suitability of explaining the relevance of the target entity e_t to the query Q.

On entity-profiles. As discussed in Section 5.1, in entity-oriented research, it is customary to use the introductory paragraph from the Wikipedia page of an entity as the entity's *profile* (short descriptions of the entity). Entity-profiles are also often built query-independently using information from Knowledge Bases such as anchor text, category links, etc., [52, 63]. However, in this work, we use a query-relevant candidate set of passages $\mathcal{D}$ to build an entity-profile and define an entity's query-relevant context $\mathcal{C}$. Hence, one might wonder: Why can't we use the introductory paragraph or the entire Wikipedia page of the entity to build the entity-profile and define the entity context $\mathcal{C}$ above?

To answer this question: We hypothesize that using a query-independent entity-profile such as the Wikipedia article, the relevant connections between the query and the target entity might be lost due to the presence of other, more popular sub-topics about the entity in the profile which are not relevant to the query. For example, some other, more popular sub-topics related to the entity "Oyster" might be "Nutrition" and "Nutrient Cycling"; however, these sub-topics are not relevant in the context of the query *Cholera*. The query-relevant candidate set of passages $\mathcal{D}$ help us to define a *query-specific* profile of the entity, thus allowing us to identify query-relevant sub-topics of the target entity.

To support our hypothesis about the benefits of using a query-specific entity profile, we replace our query-specific entity profile with the Wikipedia article of the target entity and include it in the empirical evaluation.

Constructing the query-specific entity-profile. We first retrieve a candidate set of passages $\mathcal{D}$ from an index of paragraphs with the query using BM25. We construct the query-specific entity-profile $\mathcal{D}_{e_i}$ of a target entity e_i using query-relevant candidate set of passages $\mathcal{D}$ as follows: We retain only passages $p_{e_i} \in \mathcal{D}$ that mention the target entity e_i.

5.3 Approach: Learning-To-Rank using Query-Specific Entity Profile and Entity Salience

Our approach uses a supervised combination of several features which are derived from the query-specific profile of the target entity, and the salience of the target entity in the support passage. When considering the query-specific profile, we consider both, a bag-of-entities and a bag-of-words representation of the profile. Below, we describe each individual feature, which are then treated as features and combined using Learning-To-Rank.

Notation. Henceforth, we use e_t to denote the target entity, Q to denote the query, $\mathcal{D}_{e_t}$ to denote the query-specific profile of the target entity e_t, and $p_{e_t} \in \mathcal{D}_{e_t}$ to denote a candidate support passage p_{e_t} of the target entity e_t.

5.3.1 Features Based on Query-Specific Profile

Entity Prominence (E-PROM). As discussed in Section 5.2, a passage might contain sub-topics about the target entity which are not relevant in the context of the query Q. In order to address this, we aim to model the query-relevant sub-topics of the target entity found in a candidate support passage. To this end, we treat the query-specific target entity profile $\mathcal{D}_{e_t}$ of the target entity e_t as a bag-of-entities. We use the query-relevant entities $e_x \in \mathcal{D}_{e_t}$ as a surrogate for the query-relevant sub-topics of the target entity e_t. The intuition is that since these query-relevant entities e_x have been mentioned along with the target entity e_t in a query-specific target entity profile $\mathcal{D}_{e_t}$, the entities e_x would roughly model the query-relevant sub-topics of the target entity e_t.

We derive a relevance indicator for entity-support passages based on the frequency of entities $e_x \in \mathcal{D}$ as follows:

$$P(e_x \mid e_t, Q) \quad \propto \quad \text{count}(e_x \in \mathcal{D}_{e_t}) \tag{5.1}$$

where count$(e_x \in \mathcal{D}_{e_t})$ is a function which returns the number of times a query-relevant entity e_x occurs in the query-specific profile $\mathcal{D}_{e_t}$. The intuition is that more frequently an entity e_x is mentioned along with the target entity e_t, the more probable that e_x is a query-relevant sub-topic of e_t.

Our assumption is that a candidate support passage that contains many such query-relevant sub-topics (entities) e_x of the target entity e_t is a good support passage for e_t. Hence, we model the relevance of a candidate support passage $p_{e_t} \in \mathcal{D}_{e_t}$ to the target entity e_t as follows:

$$\text{Score}_{e_t}(p_{e_t}) = \sum_{e_x \in p_{e_t}} P(e_x \mid e_t, Q) \tag{5.2}$$

This score is combined with the score of the support passage for the query to obtain the final score of the support passage:

$$\text{EPROM}(p_{e_t} \mid e_t, Q) = \lambda \cdot \text{Score}_{e_t}(p_{e_t}) + (1 - \lambda) \cdot \text{Score}_Q(p_{e_t}) \quad \lambda \in [0, 1] \tag{5.3}$$

where λ is learnt using machine learning, and $\text{Score}_Q(p_{e_t})$ is obtained from the candidate passage ranking $\mathcal{D}$ for the query.

Term Prominence (T-PROM). Alternatively, we treat the target entity profile $\mathcal{D}_{e_t}$ as a bag-of-words and use the words $t \in \mathcal{D}_{e_t}$ to rank candidate support passages p_{e_t}. Specifically, we obtain a distribution over the terms $t \in \mathcal{D}_{e_t}$ (after stop words removal) using the term frequency $\text{tf}_{p_{e_t}}(t)$ of word t in a passage $p_{e_t} \in \mathcal{D}_{e_t}$, weighted by the retrieval score $\text{Score}_Q(p_{e_t})$ of the passage p_{e_t} for the query Q as follows:

$$P(t \mid e_t, Q) \propto \sum_{p_{e_t} \in \mathcal{D}_{e_t}} \text{Score}_Q(p_{e_t}) \cdot \text{tf}_{p_{e_t}}(t) \tag{5.4}$$

where $\text{Score}_Q(p_{e_t})$ is obtained from the candidate passage ranking $\mathcal{D}$ for Q.

We then score a candidate support passage $p \in \mathcal{D}$ by accumulating the word scores of each word in the passage. Formally,

$$\text{TPROM}(p_{e_t} \mid e_t, Q) = \sum_{t \in p_{e_t}} P(t \mid e_t, Q) \tag{5.5}$$

5.3.2 Features Based On Entity Salience

As discussed in Section 5.1, a support passage must mention the target entity in a salient way, i.e., the target entity must be central to the discussion in the support passage and not just mentioned as an aside. For example, in Figure 5.2, the entity “Oyster” is salient in the left passage but not in the right. In this work, we explore if entity salience can help in the support passage retrieval task. We use the salience detection system from Ponza et al. [109][5] in our work.

Below, we describe our features which use entity salience to rank candidate support passages for a target entity and query. We denote by Salience$(e_t \mid p_{e_t})$, the salience score of the target entity e_t for a candidate support passage p_{e_t}.

Sal-ReRank. We re-rank the support passage ranking obtained using our method E-PROM in Section 5.3.1 using entity salience as follows:

$$\text{Score}(p_{e_t} \mid e_t, Q) = \lambda \cdot \text{Salience}(e_t \mid p_{e_t}) + (1 - \lambda) \cdot \text{EPROM}(p_{e_t} \mid e_t, Q) \quad \lambda \in [0, 1]$$

where λ is learnt using machine learning, and Salience$(e_t \mid p_{e_t})$ is obtained using the entity salience detection system SWAT [109]. SWAT takes a text passage as input and returns the salient entities in the passage, along with the confidence scores for each entity that shows how salient the entity is in the text.

Sal-Binary. Score$(p_{e_t} \mid e_t, Q) = 1$ if the target entity e_t is salient in the support passage p_{e_t}, and zero otherwise.

Sal-RankScoreOfPassage. Score$(p_{e_t} \mid e_t, Q)$ = Score$_Q(p_{e_t})$ if the target entity e_t is salient in the support passage p_{e_t}, and zero otherwise. Score$_Q(p_{e_t})$ is the retrieval score of the passage p_{e_t} for the query Q obtained from the candidate passage ranking $\mathcal{D}$ for the query.

Sal-SalienceScoreOfEntity. Score$(p_{e_t} \mid e_t, Q)$ = Salience$(e_t \mid p_{e_t})$ if the target entity e_t is salient in the support passage p_{e_t}, and zero otherwise.

[5] https://sobigdata.d4science.org/web/tagme/swat-api.

Sal-CombinedSaliencePassageScore. We combine the salience score of the target entity in the support passage with the score of the support passage for the query.

$$\text{Score}(p_{e_t} \mid e_t, Q) = \lambda \cdot \text{Salience}(e_t \mid p_{e_t}) + (1 - \lambda) \cdot \text{Score}_Q(p_{e_t})) \quad \lambda \in [0, 1]$$

One issue that we foresee in the use of salience for a retrieval task such as this is that many entities would not have a passage with a salient mention in the candidate set of passages $\mathcal{D}$ retrieved for the query. However, for entities which have at least one passage with a salient mention in the candidate set, our hypothesis is that entity salience would help improve retrieval performance. An initial analysis on entity salience is presented in Section .

5.4 Alternative for Evaluation: Using Wikipedia Article instead of the Query-Specific Entity-Profile

As noted in Section 5.2, previous works have used the Wikipedia article of an entity as the entity's profile. Hence, for the purpose of evaluation and comparison, we replace our query-specific target entity profile with the Wikipedia article of the target entity. Below, we describe methods which use the Wikipedia article instead of our query-specific entity profile. These methods correspond to those in Section 5.3.2.

WikiTerms. Similar to our method T-PROM described in Section 5.3.1. Here, we use the Wikipedia article $\mathcal{W}$ of the target entity e_t to find a distribution over the terms $t \in \mathcal{W}$. Specifically: $P(t \mid e_t, Q) \propto \sum_{p \in \mathcal{W}} \text{Score}_Q(p) \cdot \text{tf}_p(t)$. To calculate $\text{Score}_Q(p)$, we segment $\mathcal{W}$ into its constituent paragraphs, then index these paragraphs, and finally retrieve from this index using BM25. We use this new distribution $P(t \mid e_t, Q)$ to score the candidate support passages in Equation 5.5.

WikiEntities. Similar to E-PROM in Section 5.3.1. As in Equation 5.1, we derive a distribution $P(e_x \mid e_t, Q)$ over entities $e_x \in \mathcal{W}$. We then score a passage $p_{e_t} \in \mathcal{D}_{e_t}$ as in Equation

5.3 using the distribution $P(e_x \mid e_t, Q)$ obtained using the Wikipedia article.

5.5 Evaluation

5.5.1 Datasets

For this work, we need both, an entity ground truth and a passage ground truth. The datasets from the TREC Complex Answer Retrieval (CAR) track [33][6] contain such ground truth data and hence suitable to study this task. We use two datasets from TREC CAR to evaluate our methods. They are:

1. **BenchmarkY1-Train**. It is based on a Wikipedia dump from 2016. The Wikipedia articles are split into the outline of sections and the paragraphs contained in each section. The information about which paragraph originated from which section, and the entity links in each paragraph are retained. Each section outline is treated as a complex topic. There are 117 such sections (complex topics),

2. **BenchmarkY2-Test**. A part of this dataset is based on a Wikipedia dump from 2018 whereas the remainder is based on the Textbook Question Answering (TQA) [65] dataset which consists of questions taken from middle school science curricula. This dataset consists of 65 complex topics.

Corpus. We use the corpus of paragraphs from TREC CAR. The corpus consisting of paragraphs from the entire English Wikipedia with the entity links preserved. This corpus is constructed by collecting all paragraphs from Wikipedia, assigning unique IDs to each paragraph through SHA256 hashes on the text content (excluding links), and de-duplication through min hashing using word embedding vectors provided by GloVe.

In addition to entity links that are provided in the corpus, we create entity link annotations using WAT [104][7].

[6] `http://trec-car.cs.unh.edu`

[7] WAT has both, an entity linking system and an entity relatedness prediction system in it. These can be queried using different APIs.

Ground Truth. The TREC CAR datasets contain both passage and entity ground truth data. For *BenchmarkY1-Train*, both passage and entity ground truth were generated automatically: All paragraphs from a section in a Wikipedia page are deemed as relevant to that section, and if a page/section contains an entity link, then the link target entity is defined as relevant. The passage ground truth contains 4530 positive assessments, whereas the entity ground truth contains 13,031 positive assessments.

As mentioned above, the *BenchmarkY2-Test* dataset was constructed using pages from the Wikipedia dump of 2018. However, very few paragraphs from the Wiki-16 dump existed in the Wiki-18 dump. Moreover, the paragraph sets from Wiki-16 and TQA are disjoint. Due to this difference in the dataset construction procedure for *BenchmarkY2-Test*, the automatic ground truth extraction procedure used for constructing the passage ground truth for *BenchmarkY1-Train* could not be applied for deriving the passage ground truth for this dataset. Hence, the passage ground truth was constructed after manual assessment, and consists of 9633 positive assessments. The automatic entity ground truth construction was not affected as it does not depend on paragraph overlap. Both automatic as well as manual entity ground truth is available for *BenchmarkY2-Test* and consist of 1356 positive assessments.

Support Passage Ground Truth. We use the automatically generated ground truth (both passage and entity) for *BenchmarkY1-Train* and the manually generated ground truth (both passage and entity) for *BenchmarkY2-Test*. We derive a ground truth for entity support passage retrieval from the ground truth of relevant passages and entities provided with the data sets (article-level) as follows: Any relevant passage that contains an entity link to a relevant entity for the query is defined as relevant for the given query and entity.

5.5.2 Evaluation Paradigm

Initial Analysis on Entity Salience. We perform an initial analysis where we analyze how many relevant entities have a passage with a salient mention in the candidate set of passages retrieved for the query. We find that on average, 53% of the relevant entities for a query have no passage with a salient mention in the candidate set. For these entities, the salience indicator is not applicable. Hence, for evaluation and to answer our research questions, we restrict ourselves to entities which have at least one passage with a salience mention in the candidate set.

Candidate Passage Retrieval for Query. We use Wikipedia page titles as our queries for the initial candidate passage retrieval. To retrieve passages for a Wikipedia page title as query, we use all the section headings on the Wikipedia page to construct a boolean query of the terms in the section headings, and retrieve candidate passages with this boolean query using BM25 (Lucene default). However, any passage ranking method could be used here.

Input Entity Ranking. The input to our support passage retrieval system is a ranking of entities. We obtain an entity ranking as follows: We use an idea based on Pseudo-Relevance Feedback [70] and the Entity Context Model [26] that has been found to be a strong entity relevance indicator by prior work [18, 26, 32, 112] to obtain an entity ranking from an index of paragraphs. We represent a pseudo-relevant feedback set of paragraphs retrieved using the query as a bag-of-entities. To rank the entities in the bag, we weigh the frequency distribution of the entities by the retrieval score of the paragraphs. However, any system could be used to obtain an entity ranking here.

Wikipedia as Entity Profile. For evaluation, we use the Wikipedia article of the target entity as entity profile. The TREC CAR dataset consists of a large, unprocessed collection of Wikipedia pages. It contains all pages except those in the benchmarks. We index this unprocessed collection of Wikipedia pages using Lucene. We treat each Wikipedia page as an entity [63], and associate each entity with text that includes the Wikipedia article, as well as anchor text, names and type labels.

Machine Learning. We apply our methods to produce an entity-support passage ranking for every query-entity pair. We then treat each ranking as a feature and perform 5-fold cross validation. We use Coordinate Ascent optimized for Mean Average Precision (MAP) for this purpose.

As discussed above, our initial analysis of entity salience shows that only a few relevant entities have a passage with a salient mention in the candidate set of passages. Hence, we train our system only on entities with at least one salient passage, and evaluate only on such entities. For comparison, we also report results from training and evaluating our system on all entities.

Evaluation Metrics. In this work, as in Blanco et al. [13], we are interested in precision more than recall. This is because although there may be many passages explaining the relevance of an entity to a query, a typical user is interested in one or two of them. Moreover, the user interfaces of entity retrieval systems would typically allow for only one or two such support passages to be displayed. Hence, we use the following precision-oriented retrieval metrics to evaluate our work: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Precision at R (P@R), and Precision at 1 (P@1).

Difficulty Tests and Helps-Hurts Analysis. To analyze the extent to which a method affects the performance of our system, we perform two types of analysis:

1. Difficulty Test: We divide the query-entity pairs into different levels of difficulty according to the performance of a baseline method, with the 5% most difficult pairs for this method to the left and the 5% easiest ones to the right. We then study the performance of our methods on these different subsets of the query-entity pairs.

2. Helps-Hurts Analysis: As compared to a baseline, we calculate the number of query-entity pairs on which one of our methods improved performance (*helps*) or lowered performance (*hurt*).

5.5.3 Baselines

In this section, we describe the baselines against which we compare our methods.

Blanco et al. [13]. We re-implement the work from Blanco et al. [13] and include a learning-to-rank system of their features as a baseline. Their methods use a named entity recognizer to find entities in the candidate support sentences. We use the Stanford Named Entity Recognizer [44][8] for this purpose. Below, we give a short description of their methods which we include as features in a learning-to-rank baseline in this paper.

Given a query q and an entity e, Blanco et al. score a candidate entity support passage:

$$\text{Score}_{qe}(p) = \begin{cases} \sum_{e' \in p} E(q, e') & \text{if } e \in p \\ 0 & e \notin p \end{cases} \tag{5.6}$$

where $E(q, e')$ is an entity ranking method which scores an entity for the query.

Blanco et al. propose several alternatives for $E(q, e')$ in Equation 5.6. These are:

1. **Entity Frequency.** Number of candidate support passages mentioning an entity. This is akin to Term Frequency (TF) for terms.

2. **Entity Rarity.** Entity inverted sentence frequency to penalize very frequent entities. This is akin to Inverted Document Frequency (IDF) for terms.

3. **Combination.** Combination of Entity Frequency and Rarity as described above. This is akin to TF-IDF weighing scheme for terms.

4. **KLD.** KL-Divergence between query and collection distributions. Formally,

$$E_{KLD}(q, e) = P(e|\theta_q) \cdot \log \frac{P(e|\theta_q)}{P(e|\theta_C)}$$

where $P(e|\theta_q)$ is the proportion of the candidate passages for the query q which also mention the entity e, and $P(e|\theta_C)$ is the proportion of the passages in the entire corpus which also mention the entity e.

As in Blanco et al., we too use the results from using both, an average and a summation in Equation 5.6 with the various entity ranking methods described above.

[8] `https://nlp.stanford.edu/software/CRF-NER.html`

Other Baselines. We include the following baselines which use only the query and target entity without any other components of our approach.

1. **Frequency of relevant entity links (FreqOfRelLinks).** We rank passages for a query-entity pair by the number of relevant entities in the passage. For example, if a passage p contains entities $\{e_1, e_2\}$ and the entities $\{e_1, e_2, e_3, e_4\}$ have been retrieved for the query q, then the score of p for each of the query-entity pairs is $f_{qe_1}(p) = f_{qe_2}(p) = 2$ because the passage has two entities in common with the list retrieved for q.
2. **Compound entity-query score (CompoundQuery).** We retrieve passages using a compound query, where the query is a combination (BooleanQuery) of terms from the original query and the target entity.

5.5.4 Research Questions

In this work, we use the salience of the target entity in the support passage as an indicator of good support passage. With this, the research question we aim to answer is the following:

RQ1 To what extent is entity salience helpful in support passage retrieval?

As noted in Section 5.2, previous work on entity and document retrieval has often used the Wikipedia article of an entity as the entity's profile. However, in this work, our hypothesis is that the Wikipedia article is not suitable for support passage retrieval because it contains many sub-topics pertaining to the target entity which might be non-relevant in the context of the given query. Hence, we propose to use a query-specific entity profile. Regarding this, we study the following research question:

RQ2 What is the effect of replacing the query-specific entity-profile with the Wikipedia article of the entity?

As discussed in Section 5.5.2, for evaluation purposes, we restrict ourselves to only the subset of entities in the input entity ranking for which there is at least one passage with

Table 5.1: Main results on subset of entities with at least one passage in the candidate set in which the entity is salient. We observe from this table that our L2R system consisting of features based on our proposed query-specific entity profile, and entity salience, outperforms all the baselines, including the current state-of-the-art from Blanco et al. ▲ denotes significantly higher, at $p < 0.05$ using the paired t-test with respect to Blanco et al. [13].

		BenchmarkY1-Train				**BenchmarkY2-Test**			
		MAP	**P@R**	**MRR**	**P@1**	**MAP**	**P@R**	**MRR**	**P@1**
1	Blanco et al.	0.14	0.12	0.20	0.14	0.20	0.20	0.42	0.29
2	FreqOfEntityLinks	0.14	0.11	0.22	0.13	0.22	0.20	0.43	0.27
3	CompoundQuery	0.05	0.05	0.07	0.05	0.06	0.07	0.18	0.12
4	Ours	$\mathbf{0.31}^{\blacktriangle}$	$\mathbf{0.30}^{\blacktriangle}$	$\mathbf{0.45}^{\blacktriangle}$	$\mathbf{0.40}^{\blacktriangle}$	$\mathbf{0.47}^{\blacktriangle}$	$\mathbf{0.43}^{\blacktriangle}$	$\mathbf{0.73}^{\blacktriangle}$	$\mathbf{0.60}^{\blacktriangle}$

a salient mention in the candidate set. However, we also study the performance of our method on all entities through the following research question:

RQ3 How does the retrieval performance change, when our learning-to-rank system is trained and evaluated on all entities (salient and non-salient)?

5.6 Results and Discussions

In this section, we discuss each research question presented in Section 5.5.4. The results from using our proposed system is shown in Table 5.1. The results from the individual support passage ranking methods are shown in Table 5.2.

The results in Sections 5.6.1 and 5.6.2 are discussed with respect to the subset of entities which have at least one passage in the candidate set in which the entity is salient. In Section 5.6.3, we also discuss about what happens when we use all entities.

5.6.1 Entity Salience for Support Passage Retrieval

In this section, we discuss RQ1 by comparing our proposed system with two other systems: (1) The baseline system from Blanco et al. and, (2) A learning-to-rank system without the salience component.

Table 5.2: Performance of individual support passage ranking methods on subset of entities with at least one passage in the candidate set in which the entity is salient. $\blacktriangle$ denotes significantly higher, and $\blacktriangledown$ denotes significantly lower, at $p < 0.05$ using the paired t-test with respect to Blanco et al. [13].

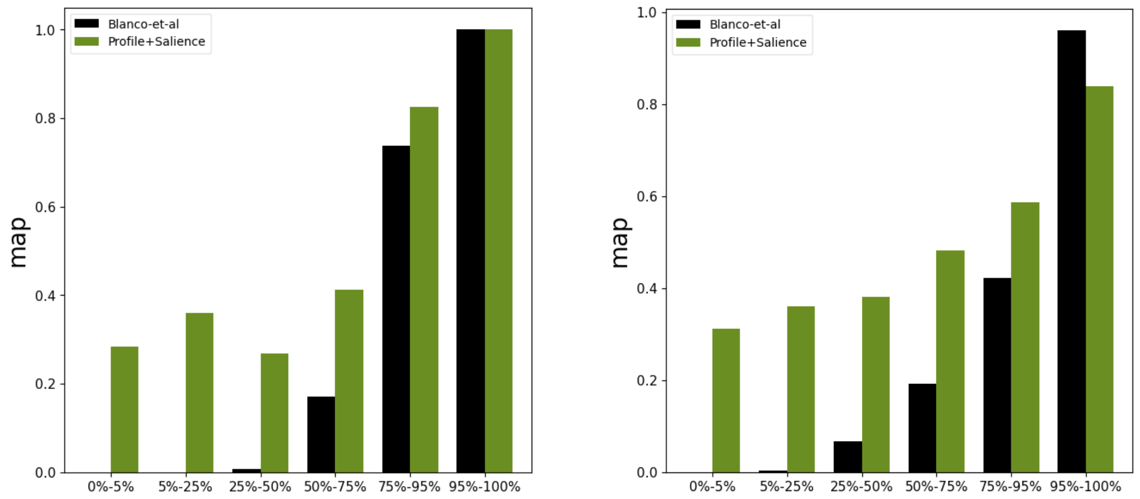
		BenchmarkY1-Train				**BenchmarkY2-Test**			
		MAP	**P@R**	**MRR**	**P@1**	**MAP**	**P@R**	**MRR**	**P@1**
1	Blanco et al. [13]	0.14	0.12	0.20	0.14	0.20	0.20	0.42	0.29
2	FreqOfEntityLinks	0.14	0.11	0.22	0.13	0.22	0.20	0.43	0.27
3	CompoundQuery	0.05	0.05	0.07	0.05	0.06	0.07	0.18	0.12
4	E-PROM	$0.24^{\blacktriangle}$	$0.22^{\blacktriangle}$	$0.33^{\blacktriangle}$	$0.26^{\blacktriangle}$	$0.34^{\blacktriangle}$	$0.31^{\blacktriangle}$	$0.54^{\blacktriangle}$	$0.36^{\blacktriangle}$
5	T-PROM	$0.23^{\blacktriangle}$	$0.20^{\blacktriangle}$	$0.32^{\blacktriangle}$	$0.23^{\blacktriangle}$	$0.38^{\blacktriangle}$	$0.34^{\blacktriangle}$	$0.64^{\blacktriangle}$	$0.51^{\blacktriangle}$
6	L2R	$0.28^{\blacktriangle}$	$0.25^{\blacktriangle}$	$0.42^{\blacktriangle}$	$0.36^{\blacktriangle}$	$0.38^{\blacktriangle}$	$0.34^{\blacktriangle}$	$0.64^{\blacktriangle}$	$0.52^{\blacktriangle}$
7	WikiTerms	$0.23^{\blacktriangle}$	$0.20^{\blacktriangle}$	$0.32^{\blacktriangle}$	$0.23^{\blacktriangle}$	$0.35^{\blacktriangle}$	$0.31^{\blacktriangle}$	$0.58^{\blacktriangle}$	$0.41^{\blacktriangle}$
8	WikiEntities	$0.12^{\blacktriangledown}$	$0.11^{\blacktriangledown}$	$0.19^{\blacktriangledown}$	$0.11^{\blacktriangledown}$	$0.28^{\blacktriangle}$	$0.28^{\blacktriangle}$	$0.51^{\blacktriangle}$	$0.36^{\blacktriangle}$
9	L2R	$0.23^{\blacktriangle}$	$0.19^{\blacktriangle}$	$0.32^{\blacktriangle}$	$0.22^{\blacktriangle}$	$0.34^{\blacktriangle}$	$0.30^{\blacktriangle}$	$0.56^{\blacktriangle}$	$0.39^{\blacktriangle}$
10	Sal-ReRank	$0.30^{\blacktriangle}$	$0.28^{\blacktriangle}$	$0.44^{\blacktriangle}$	$0.38^{\blacktriangle}$	$0.47^{\blacktriangle}$	$0.43^{\blacktriangle}$	$0.74^{\blacktriangle}$	$0.60^{\blacktriangle}$
11	Sal-Binary	$0.23^{\blacktriangle}$	$0.23^{\blacktriangle}$	$0.36^{\blacktriangle}$	$0.30^{\blacktriangle}$	$0.09^{\blacktriangledown}$	$0.12^{\blacktriangledown}$	$0.32^{\blacktriangledown}$	$0.23^{\blacktriangledown}$
12	Sal-RankScoreOfPassage	$0.25^{\blacktriangle}$	$0.26^{\blacktriangle}$	$0.41^{\blacktriangle}$	$0.36^{\blacktriangle}$	$0.12^{\blacktriangledown}$	$0.13^{\blacktriangledown}$	$0.44^{\blacktriangle}$	$0.40^{\blacktriangle}$
13	Sal-SalienceScoreOfEntity	$0.23^{\blacktriangle}$	$0.23^{\blacktriangle}$	$0.36^{\blacktriangle}$	$0.30^{\blacktriangle}$	$0.10^{\blacktriangledown}$	$0.11^{\blacktriangledown}$	$0.35^{\blacktriangledown}$	$0.27^{\blacktriangledown}$
14	Sal-CombinedSaliencePassageScore	$0.30^{\blacktriangle}$	$0.30^{\blacktriangle}$	$0.43^{\blacktriangle}$	$0.36^{\blacktriangle}$	$0.47^{\blacktriangle}$	$0.44^{\blacktriangle}$	$0.74^{\blacktriangle}$	$0.61^{\blacktriangle}$
15	L2R	$0.25^{\blacktriangle}$	$0.26^{\blacktriangle}$	$0.40^{\blacktriangle}$	$0.34^{\blacktriangle}$	$0.12^{\blacktriangle}$	$0.13^{\blacktriangle}$	$0.44^{\blacktriangle}$	$0.40^{\blacktriangle}$

Comparison with Blanco et al. From Table 5.1, we observe that on both datasets, our proposed system outperforms all baselines. It achieves statistically significant improvements over the state-of-the-art method from Blanco et al. [13].

To analyze the effects of using entity salience for support passage retrieval in more detail, we perform the difficulty test described in Section 5.5.2 and compare our proposed system with that of Blanco et al. The results are shown in Figure 5.3. From Figure 5.3, we observe that the method from Blanco et al. does well on the top (50-100%) query-entity pairs. However, it cannot find support passages for the remaining query-entity pairs. The query-entity pairs on the left are “difficult" for the method proposed by Blanco et al. However, our learning-to-rank system consistently performs well on all types of query-entity pairs (easy as well as difficult). This shows why our system outperforms the system from Blanco et al.

Using the helps-hurts analysis described in Section 5.5.2, we find that in terms of MAP, our proposed system helps improve the performance of 232 query-entity pairs on BenchmarkY1-Train, and 250 query-entity pairs on BenchmarkY2-Test.

Worked Example. As an example, in Figure 5.4, we show a query-entity pair for which



(a) BenchmarkY1-Train Results.

(b) BenchmarkY2-Test Results.

Figure 5.3: Difficulty test for MAP on the two data sets, comparing the state-of-the-art method from Blanco et al. to our learning-to-rank system consisting of features based on query-specific entity profile and entity salience. We observe that for the most difficult query-entity pairs (0-25%), our learning-to-rank system can find support passages whereas the method from Blanco et al. cannot. This helps to improve the performance.

our method successfully retrieved a support passage but the method from Blanco et al. could not. From this figure, we observe that the support passage explains the connection between the query *Pesticide* and the entity "United States Environmental Protection Agency". Moreover, the entity is salient in the support passage.

Comparison with a Learning-to-Rank System without Salience. The top part of Table 5.5 shows us that adding the salience features to the learning-to-rank system improves performance with respect to all evaluation measures on both datasets. From the difficulty test in Figure 5.5, we observe that considering only the profile features is not enough to do well on the task: In the bins marks 5-75%, entity salience helps to improve the performance and hence improves the overall results. A helps-hurts analysis shows that our system helps 139 query-entity pairs while hurting 68 on BenchmarkY1-Train. On BenchmarkY2-Test, our system helps 191 query-entity pairs and hurts 81. Hence, without the salience component, we would miss retrieving support passages for 139 query-entity pairs on BenchmarkY1-Train, and 191 on BenchmarkY2-Test.

Worked Example. As an example, in Figure 5.6, we show a query-entity pair for which

Query: Pesticide
Entity: United States Environmental Protection Agency
Support Passage:
In the United States, the Environmental Protection Agency (EPA) is responsible for regulating pesticides under the Federal Insecticide, Fungicide, and Rodenticide Act (FIFRA) and the Food Quality Protection Act (FQPA). Studies must be conducted to establish the conditions in which the material is safe to use and the effectiveness against the intended pest(s). The EPA regulates pesticides to ensure that these products do not pose adverse effects to humans or the environment. Pesticides produced before November 1984 continue to be reassessed in order to meet the current scientific and regulatory standards. All registered pesticides are reviewed every 15 years to ensure they meet the proper standards. During the registration process, a label is created. The label contains directions for proper use of the material in addition to safety restrictions. Based on acute toxicity, pesticides are assigned to a Toxicity Class.

Figure 5.4: Example query and entity with top ranked support passage found by our proposed learning-to-rank system from BenchmarkY1-Train. The method from Blanco et al. could not find a support passage for this query-entity pair, but our proposed method successfully retrieved support passages for this query-entity pair and helped to improve the performance of our system as compared to that of Blanco et al. We observe that this passage explains that the entity "United States Environmental Protection Agency" is relevant to the query *Pesticide* because the EPA regulates the pesticides in the United States. Moreover, the entity is central to the discussion in the passage.

our method successfully found a support passage but the system without the salience component could not. From this figure, we observe that the support passage explains the connection between the query *Research in lithium-ion batteries* and the entity "Massachusetts Institute of Technology". Moreover, the entity is salient in the support passage.

In Figure 5.7, we show an example of a query-entity pair for which both the systems (ours and the one without salience) found a support passage. We observe that the passage on the left does not explain why the entity "Aron Rubashkin" is relevant for the query *Agriprocessors*. Moreover, the entity is not even salient in the passage. However, the passage on the right clearly explains the relation between the entity and the query, with the entity being salient in the passage.

Take-Away. To answer **RQ1**, entity salience is a very strong indicator of support passages. Without the salience component, the support passage retrieval system cannot find

Table 5.3: Comparison of the performance of our system, when it is trained and evaluated on a subset of entities with at least one passage with a salient mention in the candidate set versus when it is trained and evaluated on all entities. We observe that when the system is trained and evaluated on only the subset, adding salience to the profile features leads to performance gains in terms of all evaluation measures. When the system is trained and evaluated on all entities, adding salience to the profile features does not affect performance. Hence, salience is a strong and useful indicator of support passages for entities for which salience is applicable. For the other entities, i.e., in the general case, it does not hurt the performance of the system using it.

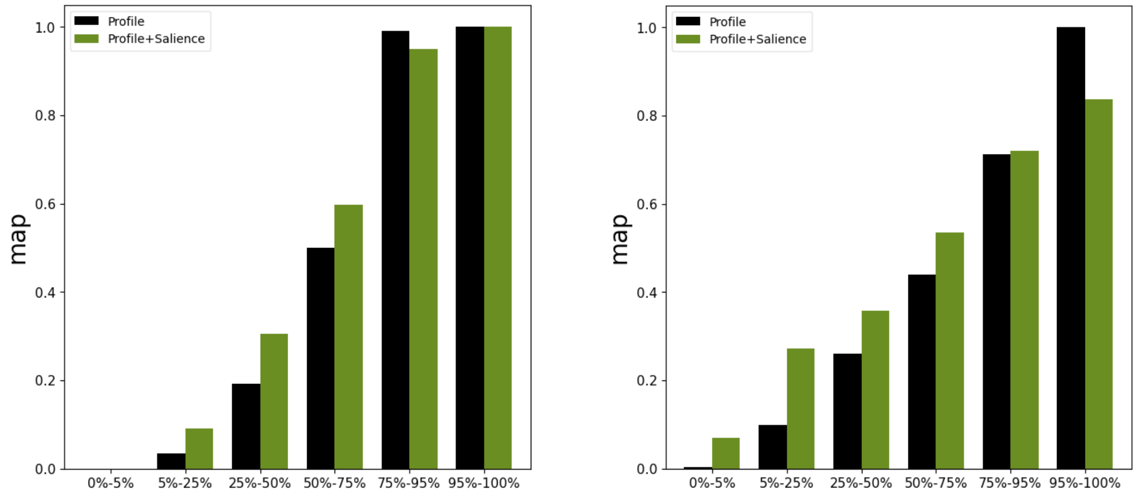
		BenchmarkY1-Train				**BenchmarkY2-Test**			
		MAP	**P@R**	**MRR**	**P@1**	**MAP**	**P@R**	**MRR**	**P@1**
Train-Subset, Evaluate-Subset	Profile	0.28	0.25	0.42	0.36	0.38	0.34	0.64	0.52
	Profile + Salience	0.31	0.30	0.45	0.40	0.47	0.43	0.73	0.60
Train-All, Evaluate-All	Profile	0.30	0.27	0.33	0.30	0.38	0.36	0.50	0.45
	Profile + Salience	0.30	0.27	0.33	0.30	0.38	0.36	0.50	0.45

meaningful passages in which the target entity is not only mentioned but also central to the discussion. As has been shown in examples above, without the salience component, the system cannot find support passages for many query-entity pairs and the performance of the system drops. From the top portion of Table 5.5, we observe that on BenchmarkY1-Train, adding salience to the system leads to an increase of 11% in terms of Mean Average Precision whereas on BenchmarkY2-Test, it leads to an improvement of 24%. For those query-entity pairs for which the system without salience does manage to find a support passage, the passage may be non-relevant and not useful as a support passage. Hence, it is important to consider the salience of the target entity while finding support passages.

5.6.2 Using Wikipedia instead of Query-Specific Entity-Profile

To answer RQ2, we show the results from replacing the query-specific entity profile with the Wikipedia article of the target entity in our learning-to-rank system in Table 5.4. We observe that the system using Wikipedia has significantly lower performance than our system which uses the query-specific entity profile.

To analyze the performance of the system using Wikipedia on the different subsets of query-entity pairs sorted by difficulty for our system, we present the results from the difficulty test in Figure 5.8. We observe that on 0-25% most difficult query-entity pairs, both



(a) BenchmarkY1-Train Results.

(b) BenchmarkY2-Test Results.

Figure 5.5: Difficulty test for MAP on the two data sets, comparing two learning-to-rank systems: one comprising of only features based on the query-specific entity profile, and other being our proposed learning-to-rank system containing features based on entity salience in addition to profile features. We observe that considering the profile features alone is not enough to achieve good results on the task. On 0-75% difficult queries on the left, the salience features help to boost the retrieval performance.

systems perform almost the same. However, for the other query-entity pairs, the system using Wikipedia does not perform as well as the system using the query-specific entity profile.

A helps-hurts analysis further shows that the system using Wikipedia instead of the query-specific entity profile helps the performance (in terms of MAP) of 63 query-entity pairs but hurts 128 of them on BenchmarkY1-Train. On BenchmarkY2-Test, the system using Wikipedia helps 74 query-entity pairs but hurts 201 of them.

Take-Away. To answer **RQ2**, replacing the query-specific entity profile with the Wikipedia article of the target entity leads to a decrease in performance. From Table 5.4, we observe that there is a drop of 6% in terms of Mean Average Precision on BenchmarkY1-Train and 23% on BenchmarkY2-Test, when the query-specific entity profile is replaced with the Wikipedia article of the target entity. This confirms our hypothesis that considering the query-specific information of the target entity is important for support passage retrieval. The issue is that a support passage for an entity must contain query-relevant topics of the

Query: Research in lithium-ion batteries
Entity: Massachusetts Institute of Technology
Support Passage:
In 2009, researchers at MIT developed a battery using genetically engineered viruses to make a more environmentally friendly battery. In 2015, another MIT group announced a flexible, puncture-resilient battery with fewer, thicker electrodes that used a semisolid aqueous suspension lithium-ion-phosphate (LFP)/lithium-titanium-phosphate (LTP) to achieve higher energy density than a conventional aqueous vanadium-redox flow battery. Using suspended particles instead of solid slabs greatly reduces the tortuosity (path length of charged particles as they move through the material).

Figure 5.6: Example query and entity with top ranked support passage found by our proposed learning-to-rank system from BenchmarkY1-Train. The learning-to-rank system containing only profile features could not find a support passage for this query-entity pair, but our proposed method successfully retrieved support passages for this query-entity pair which helped to improve the performance of our system. We observe that this passage explains that the entity "Massachusetts Institute of Technology" is relevant to the query *Research in lithium-ion batteries* because the researchers at MIT developed a new kind of lithium-ion battery. Moreover, the entity is central to the discussion in the passage.

target entity. However, the Wikipedia article contains many topics about an entity which may not be necessarily important in the context of the query, and the more important and relevant topics are lost. Considering the query-specific profile helps to narrow down on the query-relevant topics of the target entity and ignore the other, non-relevant topics.

5.6.3 Utility of Entity Salience for All Entities in General

As discussed in Section 5.5.2, we find that many entities do not have a passage in the candidate set in which the entity is salient. Hence, to draw meaningful conclusions regarding the utility of entity salience, we perform our experiments using only the subset of entities which have a passage with a salient mention in the candidate set.

To explore the effect of entity salience on the whole dataset, in this section, we present results from training and evaluating our learning-to-rank system on using all entities irrespective of whether or not they have a passage with a salient mention in the candidate set. The results are shown in Table 5.5.

From the top part of Table 5.5, we observe that for entities which have a salient passage,

Query: Agriprocessors
Entity: Aaron Rubashkin

Sholom Mordechai Rubashkin (born October 6, 1959) is an American former chief executive officer of Agriprocessors, a now-bankrupt kosher slaughterhouse and meat packing plant in Postville, Iowa, formerly owned by his father, **Aaron Rubashkin.** During his time as CEO of the plant, Agriprocessors grew into the largest kosher meat producer in the United States, but was also cited for issues involving animal treatment, food safety, environmental safety, child labor, and hiring of illegal workers.

In 1987, **Aaron Rubashkin** opened the Agriprocessors plant in Postville, Iowa and put two of his sons in charge: Sholom Rubashkin, the second youngest, as CEO; and Heshy Rubashkin, the youngest, as vice president of marketing and sales. Eventually, Agriprocessors became the United States' largest kosher slaughterhouse and meat packing plant and the only one authorized by Israel's Orthodox rabbinate to export beef to Israel. According to statistics that Rubashkin gave to Cattle Buyers Weekly, Agriprocessors' sales increased from $80 million in 1997 to $180 million in 2002. In 2002, Agriprocessors was ranked as one of the 30 biggest beef-packing plants in America.

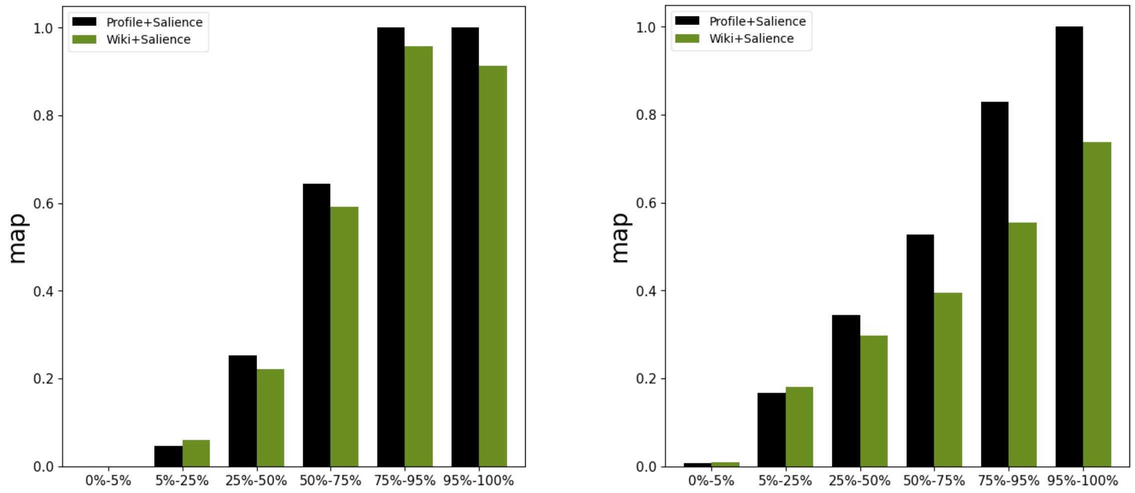
Figure 5.7: Left: Top ranked support passage found by L2R system without the salience component. Right: Top ranked support passage found by our L2R system. We observe that both passages mention the entity "Aron Rubashkin". However, the entity is not salient in the left passage whereas it is salient in the right passage. The right passage is a better explanation of why Aron Rubashkin is related to Agriprocessors: he was the founder of the plant in Iowa which became the largest meat-packing plant in the US.

Table 5.4: Results for replacing our query-specific entity profile with the Wikipedia article of the target entity in our system. These results are on the subset of entities with at least one passage in the candidate set in which the entity is salient. We observe that using Wikipedia instead of the entity profile leads to decrease in performance. $^{\blacktriangledown}$ denotes significantly lower, at $p < 0.05$ using the paired t-test with respect to our L2R (Profile + Salience) system.

		BenchmarkY1-Train				**BenchmarkY2-Test**			
		MAP	**P@R**	**MRR**	**P@1**	**MAP**	**P@R**	**MRR**	**P@1**
1	Our L2R (Profile + Salience)	0.31	0.30	0.45	0.40	0.47	0.43	0.73	0.60
2	L2R (Wikipedia + Salience)	$0.29^{\blacktriangledown}$	$0.28^{\blacktriangledown}$	$0.42^{\blacktriangledown}$	$0.36^{\blacktriangledown}$	$0.36^{\blacktriangledown}$	$0.32^{\blacktriangledown}$	$0.58^{\blacktriangledown}$	$0.43^{\blacktriangledown}$

adding salience to the system improves performance with respect to all measures. On the other hand, when we consider all entities, salience does not have any effect on the performance of a system using it, as can be observed from the bottom part of Table 5.5.

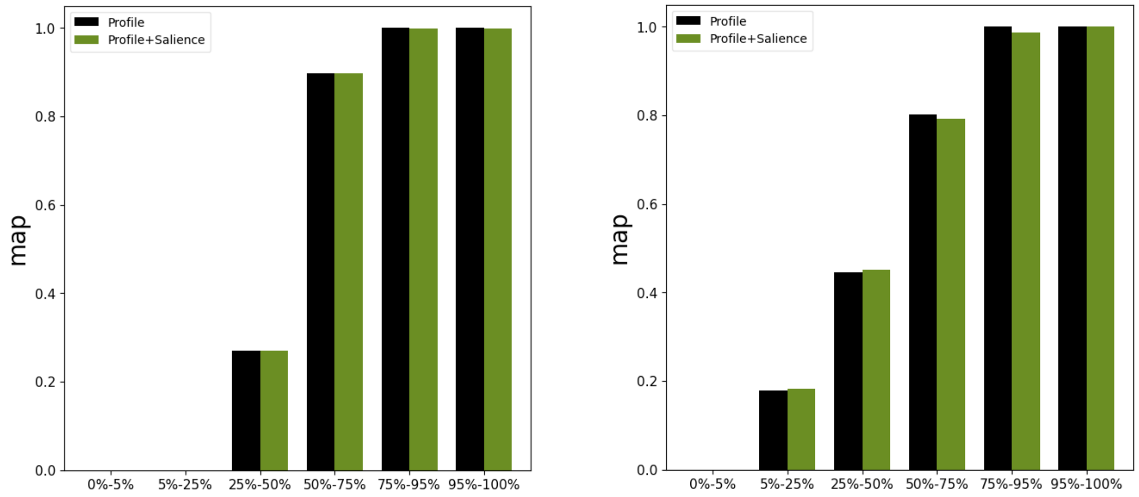
We also perform a difficulty test to analyze the performance of the following two systems, when they are trained and evaluated on all entities: our proposed learning-to-rank system, and our system without the salience component. The results are shown in Figure 5.9. We observe that the performance of our system using salience is the same as that of our system without salience on different subsets of query-entity pairs.



(a) BenchmarkY1-Train Results.

(b) BenchmarkY2-Test Results.

Figure 5.8: Difficulty test for MAP to determine the importance of using a query-specific entity profile to represent the target entity versus the Wikipedia article of the entity. The queries are divided into groups based on difficulty for our proposed system which uses the query-specific entity profile. We observe that whenever it is difficult for the system using Wikipedia to find good support passages, using the query-specific entity profile can help improve performance. This shows that information from the query-specific target entity profile is more important for support passage retrieval than that from the Wikipedia article of the target entity.



(a) BenchmarkY1-Train Results.

(b) BenchmarkY2-Test Results.

Figure 5.9: Difficulty test for MAP comparing two systems:one using the salience component and the other without it, on all entities. We observe that the salience component neither helps not hurts the performance of query-entity pairs in the different bins. This shows that although entity salience is a strong indicator of support passages for the entities for which it is applicable, in general, it does not hurt the performance of a system using it.

Table 5.5: Comparison of the performance of our system, when it is trained and evaluated on a subset of entities with at least one passage with a salient mention in the candidate set versus when it is trained and evaluated on all entities. We observe that when the system is trained and evaluated on only the subset, adding salience to the profile features leads to performance gains in terms of all evaluation measures. When the system is trained and evaluated on all entities, adding salience to the profile features does not affect performance. Hence, salience is a strong and useful indicator of support passages for entities for which salience is applicable. For the other entities, i.e., in the general case, it does not hurt the performance of the system using it.

		BenchmarkY1-Train				**BenchmarkY2-Test**			
		MAP	**P@R**	**MRR**	**P@1**	**MAP**	**P@R**	**MRR**	**P@1**
Train-Subset, Evaluate-Subset	Profile	0.28	0.25	0.42	0.36	0.38	0.34	0.64	0.52
	Profile + Salience	0.31	0.30	0.45	0.40	0.47	0.43	0.73	0.60
Train-All, Evaluate-All	Profile	0.30	0.27	0.33	0.30	0.38	0.36	0.50	0.45
	Profile + Salience	0.30	0.27	0.33	0.30	0.38	0.36	0.50	0.45

Take-Away. To answer **RQ3**, entity salience does not affect the performance when all entities are used for training and evaluation. However, from the discussion of RQ1 in Section 5.6.1, we know that salience is a very strong indicator of support passages for entities for which it is applicable. Considering these two observations together, we can say that considering salience in our system is important as it does not hurt the performance in general but helps to improve performance for entities for which salience is applicable.

5.7 Conclusion

In this work, we address the problem of entity-support passage retrieval. We present a learning-to-rank-based method which consists of two components: one component (called prominence) identifies the query-relevant sub-topics of the target entity using a query-specific entity-profile whereas the other component (called salience) considers whether the target entity is central to the discussion in the support passage and not just mentioned as an aside. Our proposed system can outperform several strong baselines in terms of several evaluation metrics and achieve new state-of-the-art results.

We find that an issue with using salience is that few entities have a passage with a salient mention in the candidate set of passages. Hence, we conduct our experiments on only the subset of entities for which there is at least one passage with a salient mention in the candidate set. Our experiments show that salience is a very strong indicator of support passages for entities to which salience is applicable. We also explore the effect of using salience for all entities and find that salience does not hurt the performance of the system in general. Our take-away from this is that salience is important for support passage retrieval; without it the entities to which salience is applicable would suffer. On the other hand, using it does not hurt the general performance of the system.

A prevalent practice for representing entities is through their Wikipedia pages. In this work, we argue that such a (static) representation of the entity is not suitable for the support passage retrieval task because support passages must contain topics related to the target entity which are also relevant in the context of the query. Hence, we use a query-specific profile of the target entity: Query-relevant text passages that mention the target entity. We explore the effects of replacing this query-specific profile with the Wikipedia page of the target entity, and find that this leads to a decline in performance. Our intuition is that the Wikipedia page contains a lot of topics/information about the target entity but only some of this is relevant in the context of the given query.

Our contribution to entity-support passage retrieval contributes to new knowledge-based information access systems. For once, it allows to construct query-specific knowledge graphs on the sub-entity level where the support passages model the knowledge base description of the entity in the context of the query. Furthermore, entity-support passages allow better information access for journalists, researchers, as well as any user who is seeking to understand fine-grained connections between entities and queries for open-domain information needs, and takes us one step closer to query-focused summarization.

CHAPTER 6

ENTITY ASPECTS FOR ENTITY RETRIEVAL

6.1 Introduction

6.1.1 Motivation

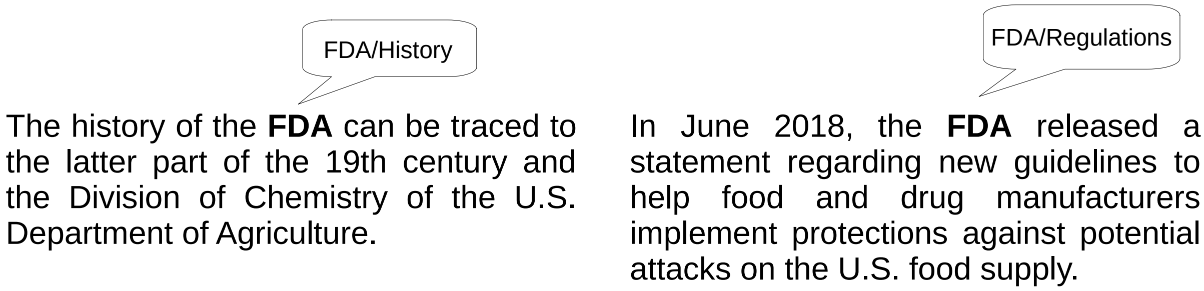
Entity-oriented Web search has become ubiquitous, with 40-70% of all web searches targeting entities [53,72,110]. Often, the information need of Web searches can be answered using a single entity, such as in conversational retrieval or factoid question answering. In contrast, in this chapter, we address the topical entity retrieval task.

Topical Entity Retrieval Task. Given a short topical keyword query such as *Antibiotic Use In Livestock*, return a ranked list of entities from a Knowledge Graph based on whether the entity must, should, or could be mentioned in an article on this topic.

We foresee this task to be useful for users seeking information on (yet) unfamiliar topics: While the information need is expressed as a short topical keyword query, the topic itself might have several facets which must be included in an article on this topic. Hence, knowing the set of relevant entities, ordered from central to side-topic might be helpful for the user.

Previous work on entity retrieval often relies on entity links for deriving indicators of entity relevance [26, 54]. Entity links are unique identifiers of entities: They help to disambiguate between the different mentions of the same entity in text. For example, using a unique entity-id, an entity link can distinguish whether a mention “FDA” in text refers to

Part of this chapter published as: Shubham Chatterjee and Laura Dietz. 2021. Entity Retrieval Using Fine-Grained Entity Aspects. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’21).* Association for Computing Machinery, New York, NY, USA. `https://doi.org/10.1145/3404835.3463035`



The history of the **FDA** can be traced to the latter part of the 19th century and the Division of Chemistry of the U.S. Department of Agriculture.

In June 2018, the **FDA** released a statement regarding new guidelines to help food and drug manufacturers implement protections against potential attacks on the U.S. food supply.

Figure 6.1: Example of entity aspect for the entity mention FDA. Entity linking can identify that the mention FDA here refers to the entity "Food and Drug Administration", the US organization and not to "Fully Differential Amplifier", the electronic instrument. However, entity linking cannot discern the *meaning* of the entity from its context. Entity aspect linking can remedy this situation using a unique aspect-id to distinguish between the different aspects of an entity. *Left:* The entity has been mentioned in the context of its history. *Right:* The entity has been mentioned in the context of its regulations.

the US organization ("Food and Drug Administration") or the electronic instrument ("Fully Differential Amplifier").

The downside of using entity links is that they can only provide limited coarse-grained information about entities in text by distinguishing between the different mentions of the same entity; however, entity links cannot provide more fine-grained information about the *meaning* of the entity. For example, knowing that "FDA" refers to the "Food and Drug Administration", entity linking cannot answer the question: Has the Food and Drug Administration been mentioned in the context of its history or regulations?

We refer to the different *meanings* of an entity in a given context as the entity's *aspects*. Our hypothesis is that entity aspects can provide better indicators of entity relevance by distinguishing between the different meanings (aspects) of an entity in the context of the query. Hence, in this work, we leverage entity aspects for the entity retrieval task and study the utility of entity aspects for entity retrieval.

6.1.2 Research Gap

As discussed in Section 6.1.1, a particular mention of an entity (e.g., FDA) may have different meanings depending on the context in which the entity has been mentioned (see Figure 6.1.

Previous work on entity retrieval often derive features for entities using entity links from a candidate set of documents retrieved with the query [26, 76]. This idea works because relevant text communicates knowledge through entities. For example, many documents on “Antibiotic Use In Livestock” would also describe the ban on antibiotics in animals by the FDA, which results in a relevance indicator for “Food and Drug Administration”. However, as discussed in Section 6.1.1, entity linking cannot distinguish between the different meanings of an entity from the entity’s context: An entity link is only a unique identifier of an entity and does not preserve any further topical information about the context in which the entity is mentioned. Entity aspect linking can remedy this situation.

Entity Aspect Linking [91, 111] is a recent information extraction task: Given a mention of an entity in a sentence, entity aspect linking refines the entity link to an entity aspect link that provides information on the context in which the entity is mentioned by indicating which aspect of the linked entity is referenced in this context. Using a unique aspect-id, entity aspect links can improve upon entity links by resolving the context (aspect) in which the entity has been mentioned in text. Hence, entity aspect linking refines an entity link with the topical semantics of the entity’s referenced aspect.

Our hypothesis is that such entity aspect links can provide additional, and perhaps better signals of the relevance of an entity for a query. Hence, in this work, we explore the extent to which such fine-grained aspects of entities can help improve entity retrieval. Analyzing entity aspect links present in a set of candidate documents allows us to significantly improve upon the current state-of-the-art in topical entity retrieval.

6.1.3 Entity Aspects Versus Entity Types

We note that entity aspects are different from entity *types*. Aspects refer to the topics in which an entity is referenced, for example, FDA in the context of its history versus FDA as a regulator; types resolve which of many roles the entity can take on, for example, US federal agency or food safety organization. Nanni et al. [91] suggest to derive a catalog of entity aspects from the top-level sections of the entity’s Wikipedia article, but other sources of aspects can also be used.

While the utility of entity types for entity retrieval is well-studied [6, 49, 63], to the best of our knowledge, we are the first to study the usefulness of entity aspects for IR tasks, such as entity retrieval.

6.1.4 Contributions

We make the following contributions through this work:

- We propose a novel entity retrieval approach which uses entity aspect links to leverage the topical context in which the entity is relevant. We outperform the previous state-of-the-art by 41%.
- We develop novel features derived from entity aspects and entity aspect linking and show that these guide our approach to more relevant and fewer non-relevant entities.
- We demonstrate that using a candidate set derived from entity-support passages (passages that are suitable to explain the relevance of an entity for a query) instead of BM25 leads to further improvements.

6.1.5 Outline

The remainder of this chapter is organized as follows. In Section 6.2, we describe our approach for using entity aspects for entity retrieval. In particular, we describe our apporach for entity ranking features using entity aspects in detail in this section. In Section 6.3 we describe the experimental methodology, followed by a discussion of the results in Section 6.4. We end the chapter with Section 6.5.

6.2 Entity Aspects for Entity Retrieval

Our work is based on the hypothesis that different mentions of an entity in a query-specific context contribute differently to determine the relevance of that entity for the query. For example, the aspect “Regulatory Programs” may be more important than “History” when determining the relevance of the entity “Food and Drug Administration” for the query “Antibiotic Use In Livestock”.

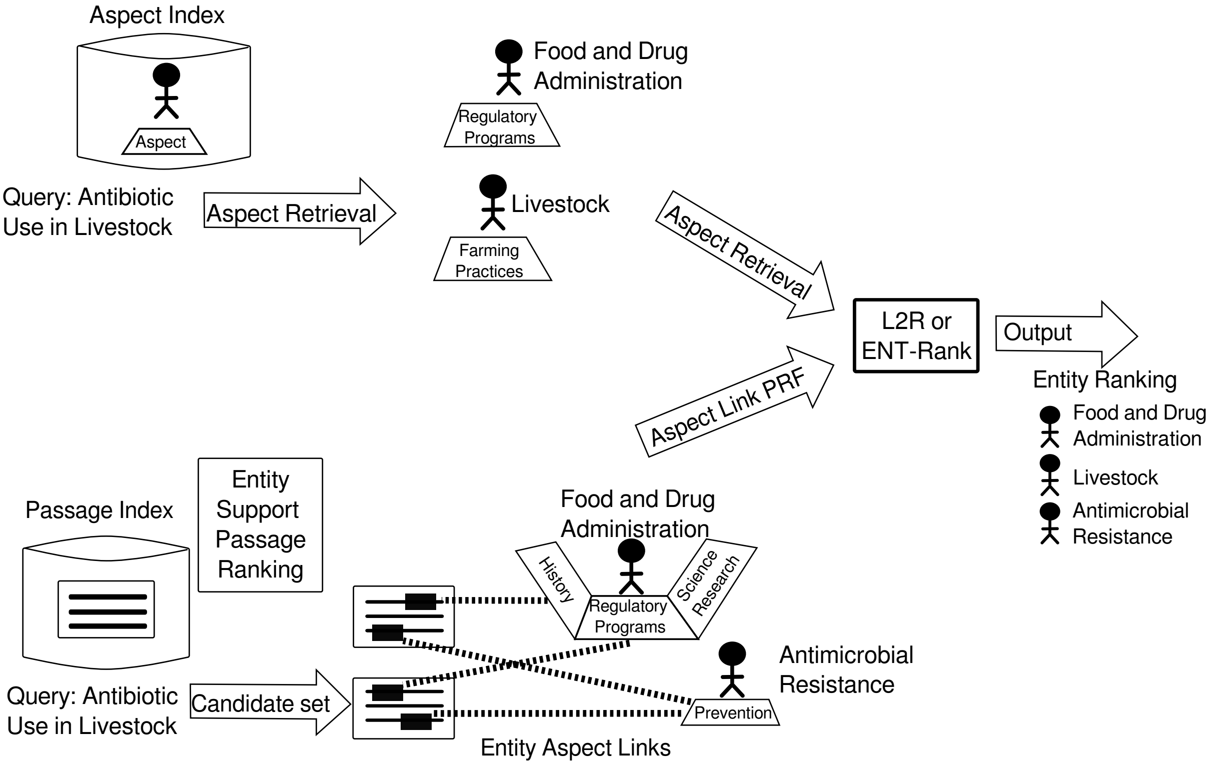


Figure 6.2: For the example query "Antibiotic Use in Livestock", we identify the relevant entity "Food and Drug Administration" as its relevant aspect "Regulatory Programs" is both retrieved from an aspect index and linked in the candidate passage set.

Each entity aspect describes the topical context in which the entity can be referred to. Entity aspects are backed by explicit semantics that are manually defined by authors of entities' Wikipedia pages. While at a first glance, it seems like a limitation to rely on Wikipedia for explicit semantics, the success of using Wikipedia in entity linking (e.g. TagMe [42]) and neural training (e.g. BERT [31]) shows that Wikipedia provides a useful repository of general-purpose knowledge that aids in many information retrieval and natural language understanding tasks.

Our approach (see Figure 6.2) towards entity ranking is to identify relevant entity-aspects, based on the assumption that only entities with relevant aspects are actually relevant for the query. At first glance, our approach of identifying relevant entity aspects seems to be addressing a more difficult problem than necessary for entity ranking. As the evaluation of entity rankings does not award credit for providing the fine-grained information about aspects, the performance improvements seem surprising. However, working with entity aspects allows us to be more specific about the topical context in which the entity is relevant. The main effect is that our predicted entity rankings contain fewer mistakes in the top of the ranking, hence obtaining significant performance improvements overall.

While the main contribution of our work is to demonstrate the benefits of using entity aspects for this task, our work draws on several ideas discussed in the research literature. Below we explain how these ideas transfer to entity aspects, and why they make this approach so successful.

Our approach relies on a corpus that is annotated with entity aspect links. In this work, we utilize the aspect catalog and aspect linker implementation provided by Ramsdell et al. [111][1] to aspect link the corpus from TREC Complex Answer Retrieval benchmark [33] which is derived from the English Wikipedia. The aspect catalog contains entity aspects derived from Wikipedia's top-level sections along with the text of the section.

[1] https://www.cs.unh.edu/~dietz/eal-dataset-2020/

6.2.1 Aspect Retrieval Features

A popular idea in entity retrieval is to create a fielded search index of entity descriptions and metadata in the knowledge base, or the full text of the entity's Wikipedia article, then use a text retrieval model to retrieve entities via their descriptions [32, 54, 92, 160]. We transfer this idea to entity aspects: We create a search index of entity aspect descriptions, comprising the entity's name (such as Food and Drug Administration), aspect's name (such as Regulatory Process), aspect's content (text of the section on Regulatory Process from the entity's Wikipedia article), and entities mentioned in the aspect's content. This information is obtained from the aspect catalog. By using multiple retrieval models, we obtain multiple aspect retrieval features which are combined with learning-to-rank.

One may argue that since aspects are derived from sections in the entity's Wikipedia article, the same information is also available when retrieving entities from a full text index of Wikipedia articles. Creating an search entry for each aspect (section) avoids the common pitfall where entities with many aspects are penalized for their diversity by the retrieval model, via document length components or L2-normalization of term vectors.

The downside of the retrieval approach is that Wikipedia articles do not always contain all relevant information about an entity. This is either because Wikipedia' policy is to only include noteworthy information, the articles are often out-of-date, or sometimes are curated to remove negative information (e.g. corporate scandals). Hence, we incorporate fall-back strategies to maximize the recall (via a PRF approach).

6.2.2 Aspect Link PRF Features

Various research in entity-oriented search are based on an idea akin to Pseudo-Relevance Feedback (PRF) [70]: After retrieving a relevant candidate set of text passages (called "feedback run"), the frequency distribution of entity links in these passages are weighted by the retrieval score of the passages to obtain a distribution of relevant entities. For example, Dalton et al. [26] uses this distribution to expand the original query with relevant entities in a manner similar to RM3 would do with words. Instead of query expansion, the distribution of relevant entities can be directly used as prediction for an entity retrieval query; this is often a very strong relevance indicator [32, 112].

In this work, we translate this idea of entity-PRF to entity aspects. Using a feedback run of passages (annotated with entity aspect links), we obtain a distribution over relevant aspects a for query q using feedback passages D akin to the expansion distribution used in RM3:

$$\text{score}(a|q) = \sum_{d \in D} \text{score}(d|q) \cdot \frac{\text{number of aspect links to } a \text{ in } d}{\text{total number of aspect links in } d}$$

Using entity aspect links instead of entity links offers access to more fine-grained topical information (“FDA/Regulatory Program” versus “FDA/History”). This helps to promote entities that are mentioned in the context of the same aspect across multiple candidate passages. Using several retrieval models (detailed in Section 6.3) and different candidate sets (detailed below), we obtain multiple aspect link PRF features which are combined with learning-to-rank.

6.2.3 Candidate Set

Our aspect-based features described above use a candidate set of passages D for the query. A common approach is to create the candidate set using the top-K documents of a BM25 ranking. However, non-relevant entities can often dominate such candidate passages, which can negatively affect the identification of relevant aspects. We avoid this by leveraging our previous work on entity-support passage retrieval [17] where the task is, given a query and a target entity, retrieve passages which best explain why the entity is relevant to the query.

We use our entity-support passage retrieval method [17]: We build an entity description consisting of passages from a query-relevant candidate set that mention the entity. These description passages are then re-ranked using query words, expansion words, and expansion entities.[2] By using entity-support passages instead of a direct BM25 ranking as

[2] In this work, this candidate set is retrieved with BM25, but the method can be adjusted to other methods as well.

candidates, we maximize the query-relevant information about each entity.

We merge all entity-support passage rankings across entities from a high-precision entity ranking.[3] We merge multiple support passage rankings by marginalizing over these entities:

$$\text{Score}(p|q) = \sum_{e_i} \text{Score}(p|e_i, q)$$

where p is a support passage for the entity e_i given the query q. The top-K of this ranking is used to build the candidate set of passages D for the query when deriving Entity Aspect Link PRF features. Such a candidate set promotes passages that are good explanations for multiple entities and avoids that the candidate ranking is dominated by a single frequently occurring entity.

6.2.4 Entity Aspect Ranking to Entity Ranking

Ultimately, we need to project rankings of entity-aspects to a ranking of entities. To this end, we consider the top-K aspects, then aggregate multiple aspects of the same entity either by sum or max. Empirically we choose $K = 100$. A separate entity ranking is obtained per aspect feature.

6.2.5 Entity Features

In addition to the entity ranking features derived using query-specific entity embeddings from BERT, we include various other entity relevance features used in previous work. To this end, we use a Wikipedia dump from 2016, and a corpus of English Wikipedia paragraphs provided with the TREC Complex Answer Retrieval dataset to create a search index representing each of the following:

- **Page.** Full-text of the Wikipedia page, including the title, headings, and the paragraphs.

[3] We retrieve entities via the lead text of their Wikipedia articles using BM25.

- **Entity.** Knowledge Graph representation of all entities, including the name of the entity and the lead text of the entity's Wikipedia page.

- **Aspect.** Top-level sections of a Wikipedia page, including the page title, section heading, and content, which includes the full-text of the section, and the entity links therein.

- **Paragraph.** Paragraphs from the corpus with full-text and entity links.

Using a keyword query and each index above, we produce entity rankings using the retrieval models described in Section 6.3.2.

We also use an idea based on Pseudo-Relevance Feedback [70] and the Entity Context Model [26] that has been found to be a strong entity relevance indicator by prior work [18, 26, 32, 112] to obtain an entity ranking from an index of paragraphs. We represent a pseudo-relevant feedback set of paragraphs retrieved using the query as a bag-of-entities. To rank the entities in the bag, we weigh the frequency distribution of the entities by the retrieval score of the paragraphs.

6.2.6 Combinations and Learning-To-Rank

We find that a strong method is a combination of aspect retrieval and aspect link PRF. Hence, we filter the aspect ranking obtained using aspect retrieval features, and only retain aspects that are linked in passages from the candidate set (either using BM25 or the support passage ranking).

We also include an aggregate feature via reciprocal rank aggregation on our aspect link features into our learning-to-rank system. Reciprocal rank aggregation is an unsupervised rank aggregation method that has been found to be a strong relevance indicator [32]: All distinct items d across all rankings R are assigned a new aggregated rank score from reciprocal ranks $\frac{1}{\text{rank}(d)}$.

We use a Learning-to-Rank[4] approach to train an ideal weighed combination of all features. We also extend the features of the original ENT-Rank approach with our entity-

[4] `https://www.cs.unh.edu/~dietz/rank-lips/`

aspect features to demonstrate the potential for further performance improvements to both the previous state-of-the-art results, as well as an improvement over learning-to-rank. In both cases, we optimize for Mean Average Precision using coordinate ascent with Z-score normalization.

6.3 Evaluation

6.3.1 Datasets

TREC Complex Answer Retrieval (CAR). The entity retrieval task of the TREC Complex Answer Retrieval (CAR) [33][5] track offers a suitable benchmark to study the topical entity retrieval task: Given a topical keyword query such as *Antibiotic Use In Livestock*, the entity retrieval task is to return a ranked list of entities based on whether the entity must, should, or could be mentioned in an article on this topic. The CAR dataset contains both manual and automatic entity ground truth, as well as an entity linked corpus consisting of paragraphs from the entire English Wikipedia. The automatic ground truth is constructed synthetically: all entities on the Wikipedia page corresponding to the query are relevant. The manual ground truth was constructed after a manual assessment conducted by NIST using pool-based evaluation. We use two subsets from the TREC CAR v2.1 data release:

- **BenchmarkY1-Train** based on a Wikipedia dump from 2016. The ground truth is automatic with 117 page-level queries, and 3,031 positive entity assessments.
- **BenchmarkY2-Test** based partly on a Wikipedia dump from 2018 and partly on the Textbook Question Answering [65] dataset (questions from middle school science curriculum). The ground truth is manual. There are 65 page-level queries with 3173 positive entity assessments, and 271 section-level queries with 1356 positive assessments.

The CAR dataset also contains a large collection of Wikipedia pages from 2016 in an easily parsable format (*unprocessedAllButBenchmark*). Query pages are excluded. We use this collection to create the page index representation used in Section 6.2.5.

[5] `http://trec-car.cs.unh.edu`

DBpedia-Entity v2. Although the focus of this dissertation is on topical queries, additionally, we also evaluate the efficacy of our approach on the DBpedia-Entity v2 [55][6] dataset. DBpedia-Entity v2 is a collection of queries collected from several established entity retrieval benchmarking campaigns. It uses the DBpedia knowledge base (October 2015). The dataset contains the following categories of queries:

- **SemSearch ES** consisting of named entity queries, e.g., *brooklyn bridge*.
- **INEX-LD** consisting of IR-style keyword queries, e.g., *electronic music genres*.
- **List Search** consisting of queries which seek a list of entities, e.g., *Professional sports teams in Philadelphia*.
- **QALD-2** consisting of natural language queries, e.g., *Who owns Aldi?*.

Since we use the paragraphs, entity links, sections, etc. from Wikipedia, we use the version[7] of DBpedia-Entity v2 projected onto the Wikipedia dump from TREC CAR v2.1. Since our methods are not included in the assessment pool, we remove the unjudged entities retrieved by our methods to enable a fair comparison.

6.3.2 Evaluation Paradigm

Evaluation Metrics. Mean Average Precision (MAP), Precision at R (P@R), and Normalized Discounted Cumulative Gain at 100 (NDCG@100). We conduct significance testing using paired-t-tests.

Learning-To-Rank We perform list-wise Learning-To-Rank (LTR) using Coordinate Ascent optimized for MAP. We use 5-fold cross-validation for fine-tuning the BERT model as well as training the LTR model on both TREC CAR and DBpedia-Entity v2.[8]

[6] https://github.com/iai-group/DBpedia-Entity

[7] https://github.com/TREMA-UNH/DBpediaV2-entity-CAR

[8] The different subsets of queries available in the DBpedia-Entity v2 collection were merged for training.

Feature Generating Retrieval Models. We produce entity rankings using the following retrieval models: (1) BM25, and (2) Query Likelihood with Dirichlet Smoothing ($\mu = 1500$), both with and without RM3-style query expansion.

6.3.3 *Baselines*

We include the following entity ranking systems as baselines:

1. **CatalogRetrieval.** We index the aspect catalog, and directly retrieve aspects from this index with the query using BM25 without any other components of our approach.
2. **GEEER [50].** The entity retrieval system from Gerritse et al. (described in Section 3.2) using Wikipedia2Vec [155] to re-rank entities.
3. **GEEER-BERT.** Same as GEEER but using BERT [106] instead of Wikipedia2Vec. We use the *name* of the Wikipedia page of the entity to embed the entity using BERT.
4. **ENT-Rank [32]** A Learning-To-Rank model that uses entity, neighbors, and text features.
5. **BM2F-CA.** Best-performing system on the DBpedia-Entity v2 dataset provided by the creators.
6. **UNH-e-L2R.** This is the best performing official submission (using BenchmarkY2-Test) to the entity retrieval track in TREC CAR Year 2. Results taken from the TREC CAR submission, which are not available for BenchmarkY1-Train.

6.3.4 *Research Questions*

We address the following research questions in this work:

RQ1 To what extent do entity apect-based features improve entity retrieval performance?

RQ2 Why do entity aspect-based features help improve entity retrieval performance?

RQ3 Is it worthwhile dedicating effort in analyzing the entity aspect links present in a candidate paragraph ranking derived from an entity-support passage ranking?

6.4 Results and Discussions

We conduct **page-level experiments** using BenchmarkY1-Train from the CAR dataset. The results are shown in Table 6.1. As the official CAR results are on title-heading queries, additionally, we also conduct **section-level experiments** using BenchmarkY2-Test. The results are shown in Table 6.2. The results on DBpedia-Entity v2 is shown in Table 6.3.

Below, we discuss the research questions outlined in Section 6.3.4 with respect to the page-level experiments using CAR BenchmarkY1-Train. We use the query *Antibiotic Use In Livestock* as an illustrative example throughout our discussions.

6.4.1 Overall Results

Page-Level Results on CAR BenchmarkY1-Train. The overall results from our page-level experiments on CAR BenchmarkY1-Train is shown in Table 6.1. We observe that our Learning-To-Rank using features based on entity aspects (LTR-ASP) outperforms all baselines in terms of all evaluation measures. In particular, LTR-ASP obtains an improvement of 56% over ENT-Rank [32], the current state-of-the-art system on this benchmark: LTR-ASP achieves $MAP = 0.50$ whereas ENT-Rank achieves $MAP = 0.32$. Further, we also outperform the neural re-ranking methods based on BERT and Wikipedia2Vec by a large margin.

Section-Level Results on CAR BenchmarkY2-Test. The overall results from our section-level experiments on CAR BenchmarkY2-Test is shown in Table 6.2. We make similar observations as those in Table 6.1. For example, our system LTR-ASP achieves an overall $MAP = 0.45$ whereas ENT-Rank achieves an overall $MAP = 0.32$: This is an improvement of 41% over ENT-Rank. LTR-ASP also outpeforms the best system from the official TREC CAR submissions (UNH-e-L2R) by 41% in terms of MAP.

Results on DBpedia-Entity v2. The overall results on DBpedia-Entity v2 is shown in Table 6.2. Here, we notice that our system LTR-ASP outperforms the best entity ranking system BM25F-CA on this dataset on all benchmarks in terms of all evaluation measures.

Table 6.1: Results on BenchmarkY1-Train page-level using automatic ground truth. $^\blacktriangle$ denotes significant improvement and $^\blacktriangledown$ denotes significant deterioration compared to ENT-Rank (denoted $\star$) using a paired-t-test at $p < 0.05$.

	MAP	P@R	NDCG@100
ENT-Rank* [32]	$0.32^\star$	$0.36^\star$	$0.46^\star$
GEEER [50]	$0.10^\blacktriangledown$	$0.16^\blacktriangledown$	$0.26^\blacktriangledown$
GEEER-BERT	$0.21^\blacktriangledown$	$0.28^\blacktriangledown$	$0.43^\blacktriangledown$
CatalogRetrieval	$0.04^\blacktriangledown$	$0.09^\blacktriangledown$	$0.15^\blacktriangledown$
LTR-ASP (Ours)	$\mathbf{0.50}^\blacktriangle$	$\mathbf{0.50}^\blacktriangle$	$\mathbf{0.63}^\blacktriangle$

Overall, BM25F-CA obtains $MAP = 0.45$ whereas LTR-ASP obtains $MAP = 0.49$, an improvement of 9%.

We also observe that LTR-ASP performs better than ENT-Rank for SemSearch_ES and INEX_LD queries; on the other benchmarks, LTR-ASP is either second or tied with ENT-Rank. This makes sense because both SemSearch_ES and INEX_LD consist of IR-style keyword queries which are very similar to the topical keyword queries from CAR. As our approach is geared towards such keyword queries, it is natural that our results also reflect this.

6.4.2 Entity Aspects for Entity Retrieval

To analyze the extent to which entity aspect-based features help imporve entity retrieval performance (RQ1), we analyze our results on BenchmarkY1-Train. We perform an ablation study where we divide our features into three subsets: aspect-based features using BM25 candidate passages, aspect-based features using entity-support passages, and other entity features. The results are shown in Table 6.4.

From Table 6.4, we observe that aspect features alone achieve $MAP = 0.42$ whereas entity features alone achieve $MAP = 0.37$. However, a combination of the three achieves the maximum improvement. The aspect-based features improve performance by 35% over the entity features.

Take-Away. Regarding **RQ1**, the ablation study shows that entity aspect-based features are important for the entity retrieval task: The aspect features help to improve performance

Table 6.2: Results on BenchmarkY2-Test (separated by its subsets on Wikipedia and TQA) section-level using the manual ground truth. $\blacktriangle$ denotes significant improvement and $\blacktriangledown$ denotes significant deterioration compared to $\star$.

All	MAP	P@R	NDCG@100
ENT-Rank* [32]	$0.32^{\star}$	$0.32^{\star}$	$0.52^{\star}$
GEEER [50]	$0.22^{\blacktriangledown}$	$0.24^{\blacktriangledown}$	$0.40^{\blacktriangledown}$
GEEER-BERT	$0.15^{\blacktriangledown}$	$0.17^{\blacktriangledown}$	$0.34^{\blacktriangledown}$
CatalogRetrieval	$0.14^{\blacktriangledown}$	$0.18^{\blacktriangledown}$	$0.27^{\blacktriangledown}$
CAR Rank 1: UNH-e-L2R	0.31	0.31	0.51
LTR-ASP (Ours)	$\mathbf{0.45}^{\blacktriangle}$	$\mathbf{0.45}^{\blacktriangle}$	$\mathbf{0.62}^{\blacktriangle}$

Textbook Question Answering [65]			
ENT-Rank* [32]	$0.38^{\star}$	$0.38^{\star}$	$0.61^{\star}$
GEEER [50]	$0.29^{\blacktriangledown}$	$0.32^{\blacktriangledown}$	$0.53^{\blacktriangledown}$
GEEER-BERT	$0.16^{\blacktriangledown}$	$0.18^{\blacktriangledown}$	$0.41^{\blacktriangledown}$
CatalogRetrieval	$0.17^{\blacktriangledown}$	$0.22^{\blacktriangledown}$	$0.33^{\blacktriangledown}$
CAR Rank 1: UNH-e-L2R	0.38	0.39	0.61
LTR-ASP (Ours)	$\mathbf{0.53}^{\blacktriangle}$	$\mathbf{0.53}^{\blacktriangle}$	$\mathbf{0.73}^{\blacktriangle}$

Wikipedia			
ENT-Rank* [32]	$0.27^{\star}$	$0.26^{\star}$	$0.43^{\star}$
GEEER [50]	$0.15^{\blacktriangledown}$	$0.16^{\blacktriangledown}$	$0.28^{\blacktriangledown}$
GEEER-BERT	$0.13^{\blacktriangledown}$	$0.15^{\blacktriangledown}$	$0.27^{\blacktriangledown}$
CatalogRetrieval	$0.13^{\blacktriangledown}$	$0.14^{\blacktriangledown}$	$0.22^{\blacktriangledown}$
CAR Rank 1: UNH-e-L2R	0.25	0.24	0.41
LTR-ASP (Ours)	$\mathbf{0.38}^{\blacktriangle}$	$\mathbf{0.38}^{\blacktriangle}$	$\mathbf{0.52}^{\blacktriangle}$

by 35% over the other entity features. We discuss this further in Section 6.4.3.

6.4.3 Importance of Entity Aspects

Query-Level Analysis. To better understand the benefits of using aspect-based features, we analyze which queries are most helped or hurt by the aspect features. To this end, we divide the queries into different levels of difficulty according to the performance of the entity features, with the 5% most difficult queries for the entity features to the left and the 5% easiest ones to the right. The results are shown in Figure 6.3.

From Figure 6.3, we observe that whenever it is difficult to perform the task using traditional entity features (e.g., bin 0–5%), aspect-based features help to improve the overall

Table 6.3: Results on DBpedia-Entity v2 (separated by different subsets). ▴ denotes significant improvement and ▾ denotes significant deterioration compared to ⋆.

All	MAP	P@R	NDCG@100
ENT-Rank⋆ [32]	$0.48^{\star}$	$0.44^{\star}$	$0.71^{\star}$
BM25F-CA [55]	0.45	$0.43^{\blacktriangle}$	0.68
GEEER [50]	$0.37^{\blacktriangledown}$	$0.38^{\blacktriangledown}$	$0.57^{\blacktriangledown}$
GEEER-BERT	$0.37^{\blacktriangledown}$	$0.38^{\blacktriangledown}$	$0.57^{\blacktriangledown}$
CatalogRetrieval	0.11	$0.14^{\blacktriangle}$	0.25
LTR-ASP (Ours)	$\mathbf{0.49}$	$\mathbf{0.45}$	$\mathbf{0.72}$

SemSearch_ES			
ENT-Rank⋆ [32]	$0.59^{\star}$	$0.50^{\star}$	$0.78^{\star}$
BM25F-CA [55]	0.61	0.55	0.78
GEEER [50]	$0.56^{\blacktriangledown}$	$0.53^{\blacktriangledown}$	$0.72^{\blacktriangledown}$
GEEER-BERT	$0.56^{\blacktriangledown}$	$0.53^{\blacktriangledown}$	$0.72^{\blacktriangledown}$
CatalogRetrieval	0.14	0.16	0.29
LTR-ASP (Ours)	$\mathbf{0.63}^{\blacktriangle}$	$\mathbf{0.57}^{\blacktriangle}$	$\mathbf{0.81}^{\blacktriangle}$

ListSearch			
ENT-Rank⋆ [32]	$\mathbf{0.49}^{\star}$	$\mathbf{0.47}^{\star}$	$\mathbf{0.74}^{\star}$
BM25F-CA [55]	0.44	$0.43^{\blacktriangle}$	0.68
GEEER [50]	$0.34^{\blacktriangledown}$	$0.38^{\blacktriangledown}$	$0.54^{\blacktriangledown}$
GEEER-BERT	$0.34^{\blacktriangledown}$	$0.38^{\blacktriangledown}$	$0.54^{\blacktriangledown}$
CatalogRetrieval	0.10	$0.14^{\blacktriangle}$	0.24
LTR-ASP (Ours)	$\mathbf{0.49}$	0.45	0.72

INDEX_LD			
ENT-Rank⋆ [32]	$0.43^{\star}$	$0.42^{\star}$	$0.70^{\star}$
BM25F-CA [55]	$0.42^{\blacktriangledown}$	$0.41^{\blacktriangle}$	$0.67^{\blacktriangledown}$
GEEER [50]	$0.34^{\blacktriangledown}$	$0.35^{\blacktriangledown}$	$0.55^{\blacktriangledown}$
GEEER-BERT	$0.34^{\blacktriangledown}$	$0.35^{\blacktriangledown}$	$0.55^{\blacktriangledown}$
CatalogRetrieval	$0.10^{\blacktriangledown}$	$0.14^{\blacktriangle}$	$0.24^{\blacktriangledown}$
LTR-ASP (Ours)	$\mathbf{0.46}^{\blacktriangle}$	$\mathbf{0.43}$	$\mathbf{0.71}$

QALD2			
ENT-Rank⋆ [32]	$\mathbf{0.40}^{\star}$	$0.37^{\star}$	$\mathbf{0.64}^{\star}$
BM25F-CA [55]	$0.37^{\blacktriangle}$	$0.36^{\blacktriangle}$	$0.46^{\blacktriangledown}$
GEEER [50]	$0.27^{\blacktriangledown}$	$0.29^{\blacktriangledown}$	$0.48^{\blacktriangledown}$
GEEER-BERT	$0.27^{\blacktriangledown}$	$0.29^{\blacktriangledown}$	$0.48^{\blacktriangledown}$
CatalogRetrieval	$0.10^{\blacktriangle}$	$0.12^{\blacktriangle}$	$0.22^{\blacktriangledown}$
LTR-ASP (Ours)	$\mathbf{0.40}$	$\mathbf{0.38}$	$\mathbf{0.64}$

Table 6.4: Results of ablation study on BenchmarkY1-Train for page-level experiments.

L2R:	Aspect		Entity			
Feedback:	BM25	Entity-Support Psg		MAP	P@R	NDCG@100
			X	0.37	0.39	0.47
	X	X		0.42	0.47	0.61
		X	X	0.48	0.49	0.62
	X		X	0.41	0.44	0.53
	X	X	X	0.50	0.50	0.62

performance of the our system LTR-ASP. We observe that the queries which were previously in the lower 5–50% percentile range with respect to the entity features obtain a MAP of 0.10, where aspect features maintain a MAP of 0.35. Even on the remaining query set, aspect features obtain a MAP of above 0.50.

Entity-Level Analysis. To understand why the aspect-based features provide a performance boost when used in a L2R setting, we further analyze our results on the entity-level. For each query in BenchmarkY1-Train, we inspect the top-100 entities in the ranking obtained by aspect-based features and entity features. We then find the number of relevant entities found by: (1) Both, (2) Only aspect-based features, and (3) Only entity features. We use the query "Antibiotic Use In Livestock" as an illustrative example here.

We find that among the 100 entities that we inspect, 16 entities are retrieved by both aspect-based features and entity features. Of these 16 common entities retrieved by both, 7 are relevant. However, aspect-based features always places these relevant entities higher in the ranking. For example, the entity "Food and Drug Administration" is placed at Rank-3 by aspect-based features but at Rank-55 by entity features.

Moreover, 11 of the remaining 84 entities retrieved by aspect-based features are relevant, but none of the remaining 84 entities retrieved by entity features are relevant. Hence, aspect-based features also retrieve more relevant entities than entity features. Some example relevant entities retrieved by aspect-based features but not by entity features are "Animal Feed", "Antibiotic Misuse", and "Antifungal".

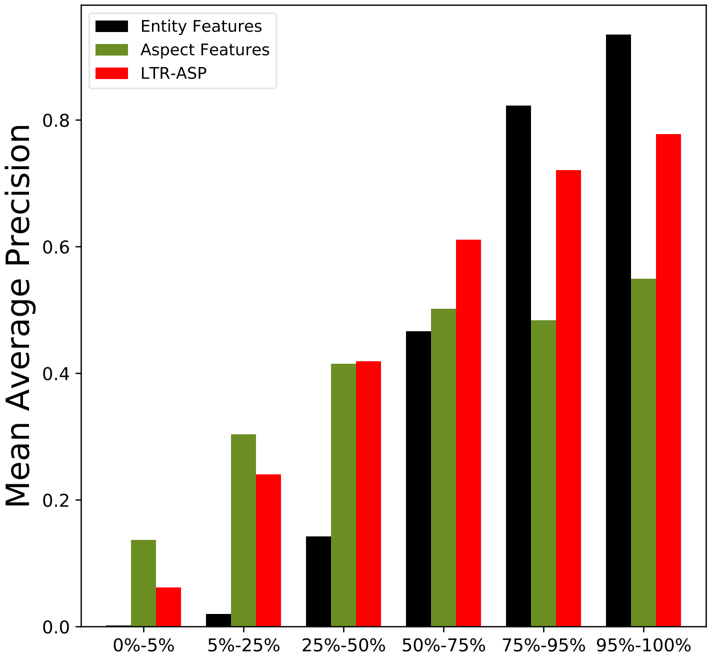


Figure 6.3: Difficulty test for MAP, comparing entity features to aspect features and LTR-ASP on BenchmarkY1-Train page-level. We notice that whenever it is difficult to perform the task using traditional entity retrieval features, aspect-based features help to improve performance.

Take-Away. The results of this analysis show that aspect-based features are more informative than traditional entity ranking features derived using term-based retrieval models. Although the non-aspect-based features perform well on their own, we receive a performance boost when these features are combined with aspect-based features (**RQ1**).

This performance boost can be attributed to the fact that aspect-based features not only promote the relevant entities higher up in the ranking, but also retrieve more relevant entities. On further feature-by-feature analysis, we find that the aspect-based features obtained using entity support passages are very strong relevance indicators and provide the maximum boost to the performance of the L2R system. We discuss this further in the next section (**RQ2**).

6.4.4 Entity-Support Passages for Deriving Aspect-based Features

To analyze the importance of the aspect-based features derived from the entity-support passages, we perform an ablation study where we first remove the aspect-based features obtained using the support passages, and then remove the aspect-based features obtained

Table 6.5: Results for analyzing why aspect-based features derived from entity-support passages are strong relevance indicators. We notice that there are more relevant entities in paragraphs at the top of the ranking obtained using entity-support passages.

	Average Percentage of Relevant Entities in	
Candidate Paragraph Ranking Obtained Using	Top-10	Top-50
Entity-Support Passage Ranking	61%	41%
BM25	47%	28%

using the BM25 candidate set from our feature mix. We then combine the remaining features using L2R. The results on BenchmarkY1-Train page-level are shown in Table 6.4.

We observe that when aspect-based features derived using entity-support passages are removed, there is a huge performance drop, from $MAP = 0.50$ obtained using all features to $MAP = 0.41$. However, when the aspect-based features obtained using BM25 candidate set are removed, the performance drops only slightly, from $MAP = 0.50$ to $MAP = 0.48$. This shows that the aspect-based features obtained using entity support passages are a stronger relevance indicator for entities.

We also find that a combination of aspect retrieval and aspect link PRF is a strong feature. Our aspect retrieval feature with BM25 achieves $MAP = 0.04$ whereas aspect link PRF with a BM25 candidate set achieves $MAP = 0.17$. However, the combination of the two achieves $MAP = 0.20$. Replacing the BM25 candidate set with a support passage candidate set for aspect link PRF in the combination achieves further improvement with $MAP = 0.40$.

The performance boost obtained when using entity aspect links from entity-support passages could be attributed to the way entity-support passages are ranked: by using other entities which frequently occur in the vicinity of the target entity in a candidate set of passages retrieved with the query. Then passages from the candidate set containing a link to many such frequently co-occurring entities are ranked higher. Hence, the entities that co-occur frequently with other query-relevant entities in query-relevant passages are emphasized. This prevents some other (non-relevant) frequent entities from dominating the ranking.

To confirm this, we collect the entities present in the top-10 and top-50 passages from two candidate sets: (1) BM25 and (2) derived using support passage ranking (SP-Derived), and analyze how many of these are relevant. The results are shown in Table 6.5. The results are shown in Table 6.5. We observe that are more relevant entities in passages at the top of the ranking in the candidate paragraph ranking derived from an entity support passage ranking.

Take-Away. Regarding **RQ3**, our approach to entity-support passage retrieval results in passages containing many query-relevant entities to be placed higher in the support passage ranking. This causes such passages to be placed higher in a passage ranking derived from such an entity-support passage ranking. As a result, the entities found in top passages are mostly relevant. Hence, it is worthwhile dedicating effort in analyzing the entity aspect links present in a candidate paragraph ranking derived from an entity-support passage ranking

6.5 Conclusion

In this chapter, we address the entity retrieval task using fine-grained aspects of entities. We demonstrate the benefits of integrating entity aspects into an entity retrieval system. Our results show that aspect-based features are more informative than traditional entity ranking features, and outperform several strong baselines, including BERT-based reranking method. When combined with other entity relevance features discussed in the literature, we obtain further improvements.

We obtain this performance boost because previously missing relevant entities are identified by aspect retrieval and aspect link PRF features when combined with L2R. Hence, relevant entities are promoted to the top of the ranking. We further find positive effects of carefully choosing the candidate set of passages for a query: significant performance improvements are obtained when replacing a BM25 candidate set with a candidate set derived from entity-support passages. Finally, our method outperforms the recent entity ranking system ENT-Rank by 41%.

In this work we demonstrate the benefits of using aspects to gain fine-grained topical information about entities using explicit semantics provided by authors of Wikipedia articles. While entity aspect linking is not widely studied in the IR community, we hope that the demonstration of its merits leads to further research in this area.

CHAPTER 7

LEARNING QUERY-SPECIFIC ENTITY REPRESENTATIONS FOR ENTITY RETRIEVAL

7.1 Introduction

7.1.1 Motivation

An important aspect of entity-oriented research pertains to the representation of entities. Commonly, the embedding of the introductory paragraph (lead text) from an entity's Wikipedia page is used as the entity's representation [77, 80, 151]. An issue with using the lead text is that it is a *static* description of the entity: Often, the lead text contains only generic information about the entity that is the same for every query and may not even be relevant for the query. For example, the entity "Food and Drug Administration" (FDA) is relevant to the topical keyword query "Genetically Modified Organism" as an organization that approved and released a kind of genetically engineered insulin; however, the lead text from the Wikipedia page of the FDA does not contain this information. In fact, the lead text has been found to be useful as an entity's description in less than 50% cases for the ClueWeb12 collection [35]. Hence, in this work, we want to use a *query-specific* textual description of an entity to encode the query-relevant information about entities.

Task. Given a query and an entity, produce a query-specific dense vector representation (embedding) of the entity.

Part of this chapter published as: Shubham Chatterjee and Laura Dietz. 2022. BERT-ER: Query-specific BERT Entity Representations for Entity Ranking. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. Association for Computing Machinery, New York, NY, USA.

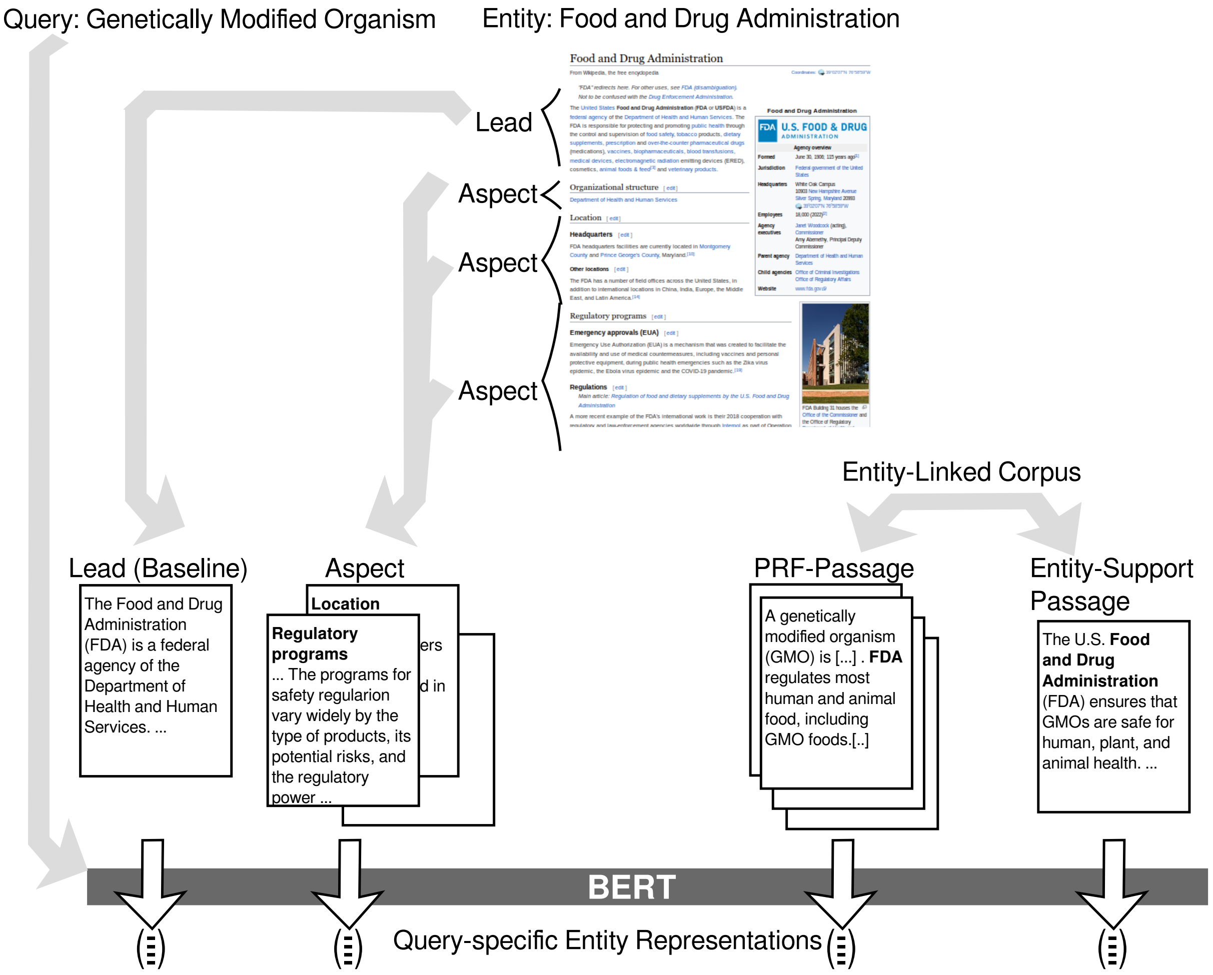

Query: Genetically Modified Organism
Entity: Food and Drug Administration
Lead
Aspect
Aspect
Aspect
Entity-Linked Corpus
Lead (Baseline)
Aspect
PRF-Passage
Entity-Support Passage
The Food and Drug Administration (FDA) is a federal agency of the Department of Health and Human Services. ...
Location
Regulatory programs
... The programs for safety regularion vary widely by the type of products, its potential risks, and the regulatory power ...
A genetically modified organism (GMO) is [...] . FDA regulates most human and animal food, including GMO foods.[..]
The U.S. Food and Drug Administration (FDA) ensures that GMOs are safe for human, plant, and animal health. ...
BERT
Query-specific Entity Representations

Figure 7.1: Example of query-specific representations for the query “Genetically Modified Organism” and entity “Food and Drug Administration”. The lead text is a generic description of the entity and has no information about the query. The PRF-passage describes the entity in the context of the query by first defining what a GMO is, then stating that the FDA regulates GMO food. As a result, the PRF-passage is a better textual description of the entity than the lead text. An issue with the PRF-passage is that the entity is not central to the discussion in the passage and the connection between FDA and GMO foods is made as a passing reference, i.e., passage is relevant to query but not to the entity. The support passage is a PRF-passage that is relevant to both the query and entity. The connection between entity and the query is central to the discussion in the support passage, and the support passage clarifies how the FDA regulates GMOs, including that the FDA allowed the use of the first genetically modified insulin (not shown in the figure). The aspect further clarifies the meaning of the entity in the context of the query – FDA is mentioned in the context of “Regulatory Program” in the support passage; the text from the aspect elaborates on this deeper query-relevant meaning of the entity.

7.1.2 Research Gap

As discussed in Section 7.1.1, the lead text from an entity’s Wikipedia page is a static textual description of the entity. As a result, the corresponding entity embedding is *static* in nature, i.e., the embedding is the same, irrespective of the query. Our intuition is that such static entity embeddings are not ideal for downstream IR tasks.

Similarly, while entity embeddings obtained using graph embedding methods [14, 73, 121, 139] encode the general semantics and knowledge of entities available in a Knowledge Graph, the embeddings are static. Recently, models such as ERNIE [159] and E-BERT [106] have been proposed in an effort to inject information from Knowledge Graphs into BERT [31]. However, these models too use a static textual description of the entity, either from Freebase or Wikipedia, resulting in static embeddings.

Static entity embeddings obtained using Wikipedia or Knowledge Graphs are easy to pre-compute and store. They have also been shown to be useful for downstream (query-independent) knowledge-driven NLP tasks such as entity linking [47, 106, 156], entity typing [103, 159] and relation classification [106, 159]. However, our intuition is that such embeddings may not be ideal for IR tasks. For example, often, a query and document are matched in the entity-space [77, 80, 91, 151] through the similarity between the embedding of the entities mentioned in the query and the document. Static entity embeddings without any knowledge of the query would be unable to identify when two entities are similar/related in the context of the query. For example, the Wikipedia page of the entity "Food and Drug Administration" does not mention the entity "Robert Swanson", yet these two entities are similar/related in the context of the query "Genetically Modified Organism" because Robert Swanson was the founder of the company that produced the first genetically engineered insulin approved for use by the Food and Drug Administration. Our hypothesis is that an entity embedding that incorporates *query-specific* knowledge about the entity would be more beneficial in a downstream IR task.

In this work, we use a *query-specific* textual description of an entity to encode the query-relevant information about entities using BERT [31]. We evaluate the impact of our query-specific BERT Entity Representations (BERT-ER) on a downstream entity ranking task: Given a keyword query, return a ranked list of entities ordered by relevance. The prevalent approach for representing entities is to produce a BERT embedding of the lead text. This approach is appealing because it is simple to implement and use; unfortunately, it leads to poor results, as we demonstrate in our experimental evaluation. We provide an approach for obtaining query-specific entity embeddings using BERT that performs much better than the prevalent approach. This improvement is achieved by incorporating query-relevant information about the entity in its representation.

In Chapter 5, we discussed about entity-support passages and in Chapter 6, we saw that using entity aspects (top-level Wikipedia sections) can help improve entity retrieval performance. Both entity-support passages and entity aspects help clarify the meaning of the entity in the context of the query. Hence, both are suitable for use as the *query-specific* description of an entity. Hence, in this chapter, explore the utility of entity-support passages and entity aspects for leaning query-specific vector representations of entities using BERT.

We explore the utility of three types of **query-specific textual descriptions** (Figure 7.1) of entities for learning query-specific entity embeddings using BERT:

- **Aspect (top-level section from Wikipedia).** We identify the relevant top-level sections from an entity's Wikipedia page, and use the text of the highest ranked section as the entity's query-specific description. Prior work [91, 111] refers to the top-level sections as an entity's *aspects*. In this work, we too refer to the top-level sections from Wikipedia as an entity's aspects. We discuss this in more detail in Section 7.2.2.

- **PRF-passage.** This is the simplest and most straightforward query-specific textual description of an entity. The approach is based in Pseudo-Relevance Feedback [70] and entity linking. We use the text of the highest ranked pseudo-relevant candidate passage that mentions an entity as the entity's query-specific textual description. We discuss this in more detail in Section 7.2.3.

- **Entity-support passage.** An entity's support passage [13,17,61] is a PRF-passage that mentions the entity and explains to a human, why an entity is relevant to a query. We use the text of the highest ranked support passage as an entity's query-specific description. We discuss this in more detail in Section 7.2.4.

7.1.3 Contributions

The novel contribution of this work is new knowledge about query-specific entity embeddings that will not only benefit the IR community but also other related research areas. In the experimental evaluation, we demonstrate the benefits of using our query-specific BERT entity embeddings using several large entity ranking benchmarks consisting of a diverse set of queries (question answering, keyword queries, list search queries, etc.).

- We obtain query-specific BERT Entity Representations (BERT-ER) by incorporating the query-relevant knowledge about an entity into its representation. This query-relevant knowledge is obtained using pseudo-relevant candidate passages, support passages, and relevant aspects (top-level sections from Wikipedia).
- Using BERT-ER in our entity ranking system, we outperform the entity ranking system that uses the BERT embedding of the lead text of entities by 13–42% on two large-scale entity ranking test collections. We also outperform systems using entity embeddings from Wikipedia2Vec [155], ERNIE [159], E-BERT [106].
- We provide a detailed empirical evaluation demonstrating that compared to the prevalent entity embedding methods, our query-specific BERT entity embeddings yield better performance for IR tasks such as entity ranking.

7.1.4 Outline

The remainder of this chapter is organized as follows. In Section 7.2, we describe our approach for obtaining the the query-specific entity representation using various query-specific textual descriptions of the entity described in Section 7.1.2. In Section 7.3, we describe our approach for entity ranking using our query-specific entity representations. In Section 7.4 we describe the experimental methodology, followed by a discussion of the results in Section 7.5. We end the chapter with Section 7.6.

7.2 Query-Specific BERT Entity Representations

Given a query and an entity, we want to produce a query-specific dense vector representation (embedding) of the entity. As discussed in Section 7.1, the prevalent approach for obtaining an entity's embedding uses the lead text from an entity's Wikipedia page. It is appealing because it is easy to implement; however, the resulting entity embeddings are static and not query-specific. Our intuition is that such static entity embeddings are not ideal for IR tasks.

In this work, we use *query-specific* entity descriptions, i.e., text that clarifies why an entity is relevant to a query. Our assumption is that such a query-specific description provides a suitable and easy-to-implement method of providing the model with query-relevant information about an entity to learn the entity's embedding. We obtain query-specific BERT Entity Representations (BERT-ER) by fine-tuning BERT for the entity ranking task.

7.2.1 Fine-tuning BERT

BERT-based neural re-ranking models such as MonoBERT and DuoBERT [95] have shown to be useful for the passage ranking task. Hence, we fine-tune a BERT model for entity ranking in two ways: (1) MonoBERT-style, point-wise model using *cross-entropy loss*[1], and (2) DuoBERT-style , pair-wise model using *margin ranking loss*.[2] The input to BERT is a sequence of query tokens and description tokens, separated by the special token *[SEP]*, and preceded by the special token *[CLS]*. We use the embedding of the *[CLS]* token obtained from the last hidden layer of BERT as the entity's query-specific embedding.

Below, we discuss the different query-specific entity descriptions used to derive query-specific entity embeddings in this work.

7.2.2 Aspects: Top-Level Wikipedia Sections

As discussed above, an entity's embedding obtained using the lead text from the entity's Wikipedia page is static and often encodes non-relevant information. Hence, we identify

[1] `https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html`

[2] `https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html`

the query-relevant information from the Wikipedia page to be used as the entity's textual description. To this end, we identify the particular top-level section from the entity's Wikipedia page that is most relevant for the query and use its text to embed the entity.

Following previous work [43, 91, 111, 114], we refer to the top-level sections from Wikipedia as *aspects*, and use a catalog of aspects provided by Ramsdell et al. [111]. To identify the most relevant top-level section (aspect) from an entity's Wikipedia page, we create a search index of aspects containing the full-text of all aspects from the catalog. We retrieve a candidate set of aspects (sections) $\mathcal{A}$ from this aspect index with the query using BM25.

An issue with directly using aspects from $\mathcal{A}$ is that many entities corresponding the aspects in $\mathcal{A}$ may not even be relevant to the query. To remedy this, we leverage prior work on entity aspect linking. Entity Aspect Linking [91, 111] refines an entity link to an entity *aspect* link by clarifying the meaning of an entity from the context in which the entity has been mentioned, for example, the entity "Food and Drug Administration" in the context of its history or regulations.

We follow a useful assumption often encountered in entity-oriented research [26, 32, 112] to further improve the quality of the candidate set of aspects $\mathcal{A}$: The entities mentioned in passages from a candidate set of passages for the query are relevant for the query. We transfer this idea to entity aspects. First, we retrieve a candidate set of passages $\mathcal{D}$ for the query using BM25, then we retain only aspects $a \in \mathcal{A}$ that are linked to atleast one passage $p \in \mathcal{D}$ to obtain a filtered candidate set of aspects $\mathcal{A}'$. We use the text of the top-ranked aspect $a_e \in \mathcal{A}'$ of an entity e as the entity's description.

The downside of the the above approach is that often, Wikipedia articles are outdated or have some (negative) information removed. As a result, they do not contain all the query-relevant information. To alleviate this problem, we explore other sources of query-specific entity descriptions (Section 7.2.3).

7.2.3 Pseudo-Relevant Candidate Passage

Alternatively, we use ideas from Pseudo-Relevance Feedback to obtain an entity's query-specific description. We use the candidate set of passages $\mathcal{D}$ for the query directly as base information: We use the text of the highest ranked passage $p_e \in \mathcal{D}$ that mentions the entity e (identified, for example, via entity links) as the entity's query-specific description.

This approach is easy to implement and based on a widely used Pseudo-Relevance Feedback technique. The downside is that although the candidate passage is relevant to the query, the entity may not be salient, i.e., *central* to the discussion in the passage, and the connection between the query and entity may be made as a passing reference. In other words, the passage may be relevant to the query but not to the entity. To overcome this limitation, we explore alternative query-specific entity descriptions. (Section 7.2.4).

7.2.4 Entity Support Passage

Prior work on entity support passage retrieval [13, 17, 61] identifies a passage that is relevant to both the query and the entity, and explains why the entity is relevant for a query. Hence, alternatively, we also use an entity's support passage as the entity's query-specific description.

We extend the ideas from previous work on entity support passage retrieval to retrieve support passages for each entity (referred to as "target entity") in a candidate entity ranking.[3] Starting with the pseudo-relevant candidate set of passages $\mathcal{D}$ obtained in Section 7.2.3, we first obtain a filtered candidate set $\mathcal{D}_e$ for a target entity e by retaining passages $p \in \mathcal{D}$ that mention the entity e. Then, we identify the k most frequently mentioned entities $e_x \in \mathcal{D}_e$. We re-rank passages $p \in \mathcal{D}_e$ for the entity e by the number of frequent entities e_x in the passage: $\text{Score}_e(p) = \sum_{e_x \in p} \text{Freq}(e_x \in \mathcal{D}_e)$, where $\text{Freq}(e_x \in \mathcal{D}_e)$ is the number of times e_x appears in $\mathcal{D}_e$. We obtain the final score of a passage $p \in \mathcal{D}_e$ by interpolating the passage's score for the entity $\text{Score}_e(p)$ with the passage's score for the query $\text{Score}_Q(p)$ (obtained from $\mathcal{D}$).

[3] We use the entity ranking obtained using the combination of Pseudo-Relevance Feedback and Entity Context Model described in Section 7.3.3

Additionally, we re-rank passages $p \in \mathcal{D}_e$ based on the salience[4] of the target entity e in the passage. Finally, we use the various support passage rankings obtained above as features to train a supervised Learning-To-Rank model, and produce one combined support passage ranking for each query and target entity. We use the text of the highest ranked support passage for each target entity as the target entity's query-specific description.

7.3 Downstream Task: Entity Ranking

7.3.1 BERT-based Entity Ranking

Given a keyword query, the entity ranking task is to return a ranked list of entities from a Knowledge Graph ordered by relevance of each entity for the query. As discussed in Section 7.2, we use MonBERT/DuoBERT-style fine-tuning to fine-tune BERT for the entity ranking task using our query-specific entity descriptions. To rank entities using BERT, the score of an entity is obtained by passing the embedding of the *[CLS]* token through a fully-connected layer trained jointly with the model.

7.3.2 List-wise Learning-To-Rank

MonoBERT is analogous to point-wise Learning-To-Rank (LTR), and DuoBERT is analogous to pair-wise LTR. However, as discussed in Section 3.2, the current state-of-the-art entity ranking models use list-wise LTR. Hence, using list-wise Learning-To-Rank, we combine the entity rankings obtained from BERT using the query-specific entity representations with other entity relevance features used in previous work [18, 26, 32] (discussed below) to obtain the final entity ranking for a query.

7.3.3 Other Entity Features

The other entity features used in this chapter have been described in Section 6.2.5.

[4] We use the salience detection system from Ponza et al. [109].

7.4 Evaluation

7.4.1 Datasets

The datasets used in the chapter have been described previously in Section 6.3.1.

7.4.2 Evaluation Paradigm

The evaluation paradigm used in the chapter has been described previously in Section 6.3.2.

7.4.3 Details of BERT for Entity Ranking

Our model is implemented in PyTorch using HuggingFace.[5]. We use the *bert-base-uncased* version of BERT. Our model is the BERT model with a fully-connected layer on top. For fine-tuning our model, we use the PyTorch implementation of the Cross-Entropy Loss and Margin Ranking Loss. The model is fine-tuned using the Adam [68][6] optimizer with a learning rate of $2e-5$ and batch size of 8. We also use a linear learning rate schedule with 1000 warm-up steps.

7.4.4 Baselines

In addition to the baselines described previously in Section 6.3.3, we include the following systems as baselines in this chapter:

1. **GEEER-ERNIE.** Same as GEEER [50] but using ERNIE [159] instead of Wikipedia2Vec [155].

2. **GEEER-E-BERT.** Same as GEEER but using E-BERT [106] instead of Wikipedia2Vec.

3. **BERT-LeadText++.** We fine-tune BERT for entity ranking using the *lead text* from an entity's Wikipedia page. The resulting entity rankings are used as features within a Learning-To-Rank system with other entity features (detailed in Section 6.2.5).

[5] https://huggingface.co/docs/transformers/model_doc/bert

[6] https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

4. **LTR-ASP [18].** Our Learning-To-Rank model from Chapter 6 that uses features based on entity aspects and entity support passages.

7.4.5 Research Questions

We address the following research questions in this chapter:

RQ1 Is it sufficient to use the lead text of an entity's Wikipedia page as the entity's description? Are query-specific entity descriptions better?

RQ2 To what extent do query-specific entity descriptions help improve entity ranking performance? What is the reason for this performance improvement?

RQ3 How do embeddings obtained using BERT-ER compare to those obtained using Wikipedia2vec for entity ranking? Which of these is better?

7.5 Results and Discussions

The overall results on CAR BenchmarkY1-Train are shown in Table 7.1, on CAR BenchmarkY2-Test in Table 7.2, and on DBpedia-Entity v2 in Table 7.3 (only best baselines shown due to lack of space). Below, we discuss the results with reference to the research questions outlined in Section 7.4.5. We use the query "Genetically Modified Organism" (GMO) as an illustrative query throughout our discussions below. In Tables 7.1 to 7.3, we refer to our entity ranking system as *BERT-ER++*. BERT-ER++ is the Learning-To-Rank combination of entity features described in Section 7.3.3 and entity rankings obtained by fine-tuning BERT using query-specific entity descriptions. In Table 7.4, BERT-ER is the Learning-To-Rank combination of all entity rankings obtained from BERT using query-specific entity descriptions (excluding the entity features described in Section 7.3.3).

7.5.1 Overall Results

From Tables 7.1 to 7.3, we observe that our entity ranking system BERT-ER++ outperforms all baselines in terms of all evaluation measures on both datasets. On the TREC CAR

Table 7.1: Results on BenchmarkY1-Train page-level using automatic ground truth. $\blacktriangle$ denotes significant improvement and $\blacktriangledown$ denotes significant deterioration compared to BERT-LeadText++ (denoted $\star$) using a paired-t-test at $p < 0.05$.

	MAP	P@R	NDCG@100
BERT-LeadText++*	$0.38^{\star}$	$0.41^{\star}$	$0.49^{\star}$
GEEER [50]	$0.15^{\blacktriangledown}$	$0.21^{\blacktriangledown}$	$0.30^{\blacktriangledown}$
GEEER-E-BERT	$0.13^{\blacktriangledown}$	$0.18^{\blacktriangledown}$	$0.26^{\blacktriangledown}$
GEEER-ERNIE	$0.14^{\blacktriangledown}$	$0.19^{\blacktriangledown}$	$0.26^{\blacktriangledown}$
GEEER-BERT	$0.14^{\blacktriangledown}$	$0.21^{\blacktriangledown}$	$0.28^{\blacktriangledown}$
ENT-Rank [32]	$0.32^{\blacktriangledown}$	$0.36^{\blacktriangledown}$	$0.46^{\blacktriangledown}$
LTR-ASP [18]	$0.49^{\blacktriangle}$	$0.50^{\blacktriangle}$	$0.63^{\blacktriangle}$
BERT-ER++	$\mathbf{0.54}^{\blacktriangle}$	$\mathbf{0.54}^{\blacktriangle}$	$\mathbf{0.66}^{\blacktriangle}$

dataset, in comparison to BERT-LeadText++, we obtain an improvement of 42% in terms of MAP ($MAP = 0.38$ to $MAP = 0.54$) on BenchmarkY1-Train in Table 7.1, and 32% in terms of MAP ($MAP = 0.25$ to $MAP = 0.33$) on BenchmarkY2-Test (All) in Table 7.2. On DBpedia-Entity v2, we obtain an improvement of 13% (overall) in terms of MAP ($MAP = 0.48$ to $MAP = 0.51$) in Table 7.3. BERT-ER (Table 7.4) and BERT-ER++ especially improve on the recall-oriented measures MAP and NDCG@100. This shows that query-specific entity descriptions are more informative and useful than the lead text of an entity's Wikipedia article that has often been used in prior work.

BERT-ER and BERT-ER++ also obtain statistically significant improvements over the entity re-ranking systems using recent and state-of-the-art entity embedding methods: Wikipedia2Vec [155], ERNIE [159], and E-BERT [106]. For example, on CAR BenchmarkY1-Train in Table 7.1, GEEER [50] using Wikipedia2Vec obtain $MAP = 0.15$, GEEER-E-BERT obtains $MAP = 0.13$, and GEEER-ERNIE obtaines $MAP = 0.14$; our system BERT-ER++ obtains $MAP = 0.54$. Similar results are observed in Tables 7.2 and 7.3.

7.5.2 Importance of Query-Specific Descriptions

To investigate why BERT-ER++ performs so well, we remove the other entity features from BERT-ER++ and analyze the results obtained by only BERT-ER. The results are shown in Table 7.4. This table shows the results of fine-tuning BERT for entity ranking using the individual query-specific entity descriptions obtained in Section 7.2 as well as a Learning-

Table 7.2: Results on BenchmarkY2-Test (separated by its subsets on Wikipedia and TQA) page-level using the manual ground truth. $\blacktriangle$ denotes significant improvement and $\blacktriangledown$ denotes significant deterioration compared to $\star$. ENT-Rank results on BenchmarkY2-Test page-level unavailable.

All	MAP	P@R	NDCG@100
BERT-LeadText++*	$0.25^{\star}$	$0.29^{\star}$	$0.44^{\star}$
GEEER [50]	$0.06^{\blacktriangledown}$	$0.11^{\blacktriangledown}$	$0.18^{\blacktriangledown}$
GEEER-E-BERT	$0.04^{\blacktriangledown}$	$0.08^{\blacktriangledown}$	$0.13^{\blacktriangledown}$
GEEER-ERNIE	$0.04^{\blacktriangledown}$	$0.08^{\blacktriangledown}$	$0.14^{\blacktriangledown}$
GEEER-BERT	$0.02^{\blacktriangledown}$	$0.07^{\blacktriangledown}$	$0.09^{\blacktriangledown}$
LTR-ASP [18]	0.24	$0.31^{\blacktriangle}$	$0.46^{\blacktriangle}$
BERT-ER++	$\mathbf{0.33}^{\blacktriangle}$	$\mathbf{0.36}^{\blacktriangle}$	$\mathbf{0.54}^{\blacktriangle}$
Textbook Question Answering [65]			
BERT-LeadText++*	$0.25^{\star}$	$0.28^{\star}$	$0.46^{\star}$
GEEER [50]	$0.06^{\blacktriangledown}$	$0.10^{\blacktriangledown}$	$0.19^{\blacktriangledown}$
GEEER-E-BERT	$0.03^{\blacktriangledown}$	$0.07^{\blacktriangledown}$	$0.12^{\blacktriangledown}$
GEEER-ERNIE	$0.03^{\blacktriangledown}$	$0.05^{\blacktriangledown}$	$0.10^{\blacktriangledown}$
GEEER-BERT	$0.01^{\blacktriangledown}$	$0.04^{\blacktriangledown}$	$0.06^{\blacktriangledown}$
LTR-ASP [18]	0.29	$0.34^{\blacktriangle}$	$0.52^{\blacktriangle}$
BERT-ER++	$\mathbf{0.33}^{\blacktriangle}$	$\mathbf{0.37}^{\blacktriangle}$	$\mathbf{0.55}^{\blacktriangle}$
Wikipedia			
BERT-LeadText++*	$0.24^{\star}$	$0.28^{\star}$	$0.40^{\star}$
GEEER [50]	$0.07^{\blacktriangledown}$	$0.12^{\blacktriangledown}$	$0.17^{\blacktriangledown}$
GEEER-E-BERT	$0.07^{\blacktriangledown}$	$0.12^{\blacktriangledown}$	$0.18^{\blacktriangledown}$
GEEER-ERNIE	$0.05^{\blacktriangledown}$	$0.09^{\blacktriangledown}$	$0.13^{\blacktriangledown}$
GEEER-BERT	$0.05^{\blacktriangledown}$	$0.12^{\blacktriangledown}$	$0.14^{\blacktriangledown}$
LTR-ASP [18]	0.29	$0.32^{\blacktriangle}$	$0.47^{\blacktriangle}$
BERT-ER++	$\mathbf{0.34}^{\blacktriangle}$	$\mathbf{0.36}^{\blacktriangle}$	$\mathbf{0.50}^{\blacktriangle}$

To-Rank combination of these (denoted as BERT-ER in the table). We use Equation 3.1 to rank entities using Wikipedia2Vec, E-BERT and ERNIE. We show results for only CAR BenchmarkY1-Train and DBpedia-Entity v2 (All).

Ablation Study. From Table 7.4, we observe that BERT-ER outperforms BERT-LeadText on both datasets. On CAR BenchmarkY1-Train, BERT-ER achieves $MAP = 0.34$ whereas BERT-LeadText achieves $MAP = 0.16$. On DBpedia-Entity v2, BERT-ER achieves $MAP = 0.22$ whereas BERT-LeadText achieves $MAP = 0.07$. We also observe that BERT-SupportPsg,

Table 7.3: Results on DBpedia-Entity v2 (separated by different subsets). $^{\blacktriangle}$ denotes significant improvement and $^{\blacktriangledown}$ denotes significant deterioration compared to $\star$. Only best baselines shown.

All	MAP	P@R	NDCG@100
BERT-LeadText++*	$0.45^{\star}$	$0.41^{\star}$	$0.68^{\star}$
BM25F-CA [55]	0.45	$0.43^{\blacktriangle}$	0.68
ENT-Rank [32]	$0.48^{\blacktriangle}$	$0.44^{\blacktriangle}$	$0.71^{\blacktriangle}$
GEEER [50]	$0.37^{\blacktriangledown}$	$0.38^{\blacktriangledown}$	$0.57^{\blacktriangledown}$
LTR-ASP [18]	$0.43^{\blacktriangledown}$	$0.39^{\blacktriangledown}$	0.68
BERT-ER++	$\mathbf{0.51}^{\blacktriangle}$	$\mathbf{0.47}^{\blacktriangle}$	$\mathbf{0.73}^{\blacktriangle}$
SemSearch_ES			
BERT-LeadText++*	$0.60^{\star}$	$0.54^{\star}$	$0.77^{\star}$
BM25F-CA [55]	0.61	0.55	0.78
ENT-Rank [32]	0.59	$0.50^{\blacktriangledown}$	0.78
GEEER [50]	$0.56^{\blacktriangledown}$	$0.53^{\blacktriangledown}$	$0.72^{\blacktriangledown}$
LTR-ASP [18]	$0.55^{\blacktriangledown}$	$0.47^{\blacktriangledown}$	$0.74^{\blacktriangledown}$
BERT-ER++	$\mathbf{0.64}^{\blacktriangle}$	$\mathbf{0.58}^{\blacktriangle}$	$\mathbf{0.81}^{\blacktriangle}$
ListSearch			
BERT-LeadText++*	$0.43^{\star}$	$0.40^{\star}$	$0.69^{\star}$
BM25F-CA [55]	0.44	$0.43^{\blacktriangle}$	0.68
ENT-Rank [32]	$0.49^{\blacktriangle}$	$0.47^{\blacktriangle}$	$0.74^{\blacktriangle}$
GEEER [50]	$0.34^{\blacktriangledown}$	$0.38^{\blacktriangledown}$	$0.54^{\blacktriangledown}$
LTR-ASP [18]	0.43	$0.41^{\blacktriangle}$	0.69
BERT-ER++	$\mathbf{0.51}^{\blacktriangle}$	$\mathbf{0.47}^{\blacktriangle}$	$\mathbf{0.74}^{\blacktriangle}$
INDEX_LD			
BERT-LeadText++*	$0.43^{\star}$	$0.40^{\star}$	$0.69^{\star}$
BM25F-CA [55]	$0.42^{\blacktriangledown}$	$0.41^{\blacktriangle}$	$0.67^{\blacktriangledown}$
ENT-Rank [32]	0.43	$0.42^{\blacktriangle}$	$0.70^{\blacktriangle}$
GEEER [50]	$0.34^{\blacktriangledown}$	$0.35^{\blacktriangledown}$	$0.55^{\blacktriangledown}$
LTR-ASP [18]	$0.41^{\blacktriangledown}$	$0.38^{\blacktriangledown}$	$0.67^{\blacktriangledown}$
BERT-ER++	$\mathbf{0.47}^{\blacktriangle}$	$\mathbf{0.44}^{\blacktriangle}$	$\mathbf{0.71}^{\blacktriangle}$
QALD2			
BERT-LeadText++*	$0.34^{\star}$	$0.32^{\star}$	$0.60^{\star}$
BM25F-CA [55]	$0.37^{\blacktriangle}$	$0.36^{\blacktriangle}$	$0.46^{\blacktriangledown}$
ENT-Rank [32]	$0.40^{\blacktriangle}$	0.37	$0.64^{\blacktriangle}$
GEEER [50]	$0.27^{\blacktriangledown}$	$0.29^{\blacktriangledown}$	$0.48^{\blacktriangledown}$
LTR-ASP [18]	$0.36^{\blacktriangle}$	0.32	$0.62^{\blacktriangle}$
BERT-ER++	$\mathbf{0.43}^{\blacktriangle}$	$\mathbf{0.39}^{\blacktriangle}$	$\mathbf{0.66}^{\blacktriangle}$

Table 7.4: Ablation study. Results on CAR BenchmarkY1-Train (Automatic) and DBpedia-Entity v2 (All) for entity ranking using different types of embeddings. $\blacktriangle$ denotes significant improvement and $\blacktriangledown$ significant deterioration compared to $\star$.

	CAR Y1-Train (Automatic)			DBpedia-Entity v2 (All)		
	MAP	P@R	NDCG@100	MAP	P@R	NDCG@100
BERT-LeadText*	$0.16^{\star}$	$0.20^{\star}$	$0.25^{\star}$	$0.07^{\star}$	$0.08^{\star}$	$0.12^{\star}$
Wikipedia2Vec [155]	$0.10^{\blacktriangledown}$	$0.16^{\blacktriangledown}$	$0.23^{\blacktriangledown}$	$0.05^{\blacktriangledown}$	$0.07^{\blacktriangledown}$	$0.10^{\blacktriangledown}$
E-BERT [106]	$0.11^{\blacktriangledown}$	$0.13^{\blacktriangledown}$	$0.19^{\blacktriangledown}$	$0.09^{\blacktriangle}$	$0.15^{\blacktriangle}$	$0.22^{\blacktriangle}$
ERNIE [159]	$0.05^{\blacktriangledown}$	$0.10^{\blacktriangledown}$	$0.14^{\blacktriangledown}$	$0.09^{\blacktriangle}$	$0.12^{\blacktriangle}$	$0.16^{\blacktriangle}$
BERT-BM25Psg	$0.06^{\blacktriangledown}$	$0.07^{\blacktriangledown}$	$0.11^{\blacktriangledown}$	$0.08^{\blacktriangle}$	$0.10^{\blacktriangle}$	$0.14^{\blacktriangle}$
BERT-SupportPsg	$0.29^{\blacktriangle}$	$0.32^{\blacktriangle}$	$0.44^{\blacktriangle}$	$0.14^{\blacktriangle}$	$0.16^{\blacktriangle}$	$0.24^{\blacktriangle}$
BERT-Aspects	$0.22^{\blacktriangle}$	$0.28^{\blacktriangle}$	$0.37^{\blacktriangle}$	$0.18^{\blacktriangle}$	$0.21^{\blacktriangle}$	$0.30^{\blacktriangle}$
BERT-ER	$\mathbf{0.34}^{\blacktriangle}$	$\mathbf{0.36}^{\blacktriangle}$	$\mathbf{0.48}^{\blacktriangle}$	$\mathbf{0.22}^{\blacktriangle}$	$\mathbf{0.23}^{\blacktriangle}$	$\mathbf{0.35}^{\blacktriangle}$

BERT-Aspects, and BERT-ER outperform Wikipedia2Vec, E-BERT and ERNIE on both datasets. This is because Wikipedia2Vec, E-BERT and ERNIE produce *query-independent* entity embeddings (embeddings have no knowledge of the query) using a query-independent textual description of an entity (often, the Freebase description or lead text). Hence, their performance on an IR task (here, entity ranking) is not good. BERT-SupportPsg and BERT-Aspects use *query-specific* entity embeddings obtained using query-specific entity descriptions. As a result, BERT-ER (combining BERT-SupportPsg, BERT-Aspects, and BERT-BM25Psg) is able to differentiate between relevant and non-relevant entities better and outperforms all other methods.

Lead Text Versus Query-Specific Descriptions. To investigate the source of performance improvements due to query-specific entity descriptions, we analyze the results at the query-level by dividing the queries into different levels of difficulty according to the performance (MAP) of BERT-LeadText. We put the 5% most difficult queries for BERT-LeadText to the left and the 5% easiest ones to the right. Below, we discuss the results only with respect to CAR BenchmarkY1-Train but similar results are obtained on the other benchmarks.

From Figure 7.2, we observe that BERT-SupportPsg, BERT-Aspects, and BERT-BM25Psg perform well on the "difficult" queries (e.g., bins 0-50%) on which BERT-LeadText performs

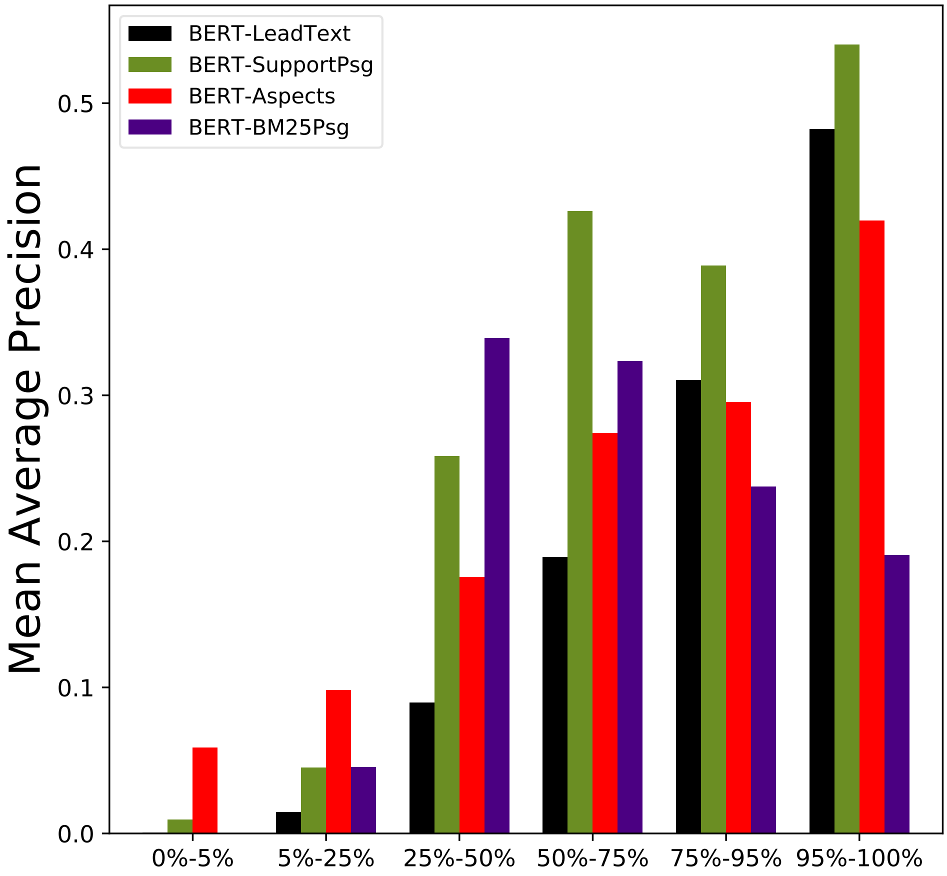
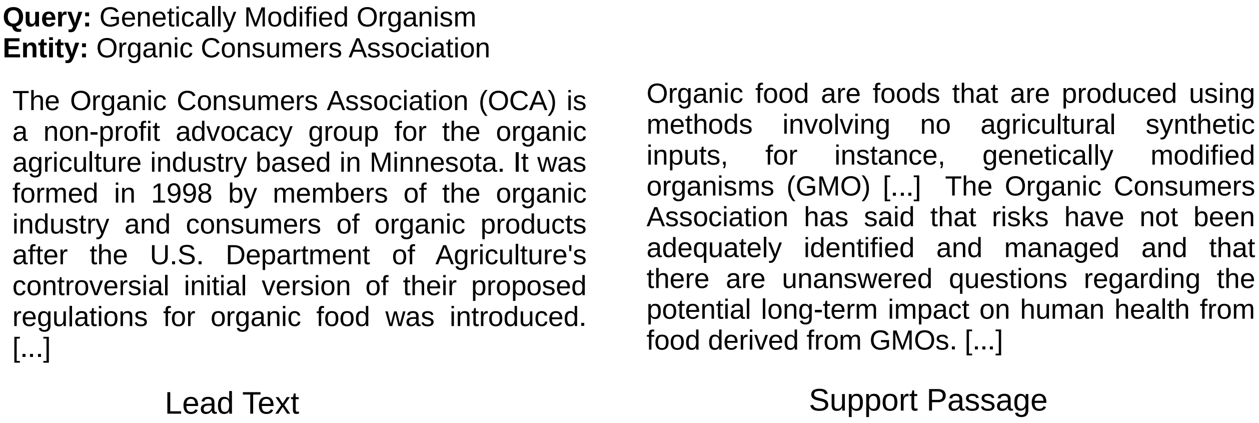


Figure 7.2: Difficulty test for MAP on CAR BenchmarkY1-Train, comparing entity rankings obtained by fine-tuning BERT using different query-specific entity descriptions. Baseline: BERT-LeadText. 5% most difficult queries for BERT-LeadText to the left and the 5% easiest ones to the right. Performance reported as macro-averages across queries. For the difficult queries (0-50%), relevant entities are found using BM25 passages, entity support passages, entity aspects. Hence, our entity ranking system outperforms several baselines.

poorly. BERT-SupportPsg is always better than BERT-LeadText, even for queries where BERT-LeadText performs the best (bin 95–100%). BERT-Aspects are better than BERT-LeadText on 75% of the queries. We also notice that BERT-BM25Psg is complementary to BERT-LeadText: When the performance of BERT-LeadText is low, BERT-BM25Psg performs well, for example, in bin 25–50%, and vice-versa.

We find that BERT-SupportPsg improves performance (*helps*) on 92 queries, BERT-Aspects helps 95 queries, and BERT-BM25Psg helps 18 queries. On inspecting the top-100 entities for some queries that are helped, we find that compared to BERT-LeadText, BERT-SupportPsg, BERT-Aspects, and BERT-BM25Psg place relevant entities higher in the ranking. For example, BERT-LeadText places the relevant entity “Organic Consumers



Query: Genetically Modified Organism
Entity: Organic Consumers Association

The Organic Consumers Association (OCA) is a non-profit advocacy group for the organic agriculture industry based in Minnesota. It was formed in 1998 by members of the organic industry and consumers of organic products after the U.S. Department of Agriculture's controversial initial version of their proposed regulations for organic food was introduced. [...]

Lead Text

Organic food are foods that are produced using methods involving no agricultural synthetic inputs, for instance, genetically modified organisms (GMO) [...] The Organic Consumers Association has said that risks have not been adequately identified and managed and that there are unanswered questions regarding the potential long-term impact on human health from food derived from GMOs. [...]

Support Passage

Figure 7.3: Example query and entity with description. Left: Lead text. The passage is a generic description of the entity and does not elaborate upon the connection between the query and entity. Right: Support passage. The passage is relevant to the query and elaborates that the entity "Organic Consumers Association" is relevant to GMOs because it regulates GMO food. This query-relevant knowledge helps BERT-SupportPsg learn that the entity is relevant for the query and promotes it up the ranking, from rank-57 placed by BERT-LeadText to rank-13.

Association" at rank 57 whereas BERT-SupportPsg places it at rank 13 (see Figure 7.3). By promoting relevant entities higher up in the ranking, query-specific entity descriptions help to improve the precision at the top of the ranking. Moreover, we are able to improve performance on the "difficult" queries for BERT-LeadText using query-specific entity descriptions.

BM25 Passage as Description. From Table 7.4, we observe that on CAR BenchmarkY1-Train, BERT-BM25Psg obtains $MAP = 0.06$ whereas BERT-SupportPsg obtains $MAP = 0.29$. Using the difficulty test above, we find that BERT-SupportPsg obtains $MAP = 0.30$ on the lower 0–50% (difficult) queries where BERT-BM25Psg obtains $MAP = 0.15$. This shows that using an entity's support passage as its description is better than using a query-relevant BM25-passage that mentions the entity. As discussed in Section 7.2.4, this is because sometimes, the entity may not be salient to the discussion in the BM25-passage, and the connection between the query and entity may be made as a passing reference, i.e., although the passage is relevant to the query, it is non-relevant for the entity (see Figure 7.1 for example). A support passage addresses this issue because the support passage retrieval method only considers passages which are relevant to the query and mention the entity in a salient manner.

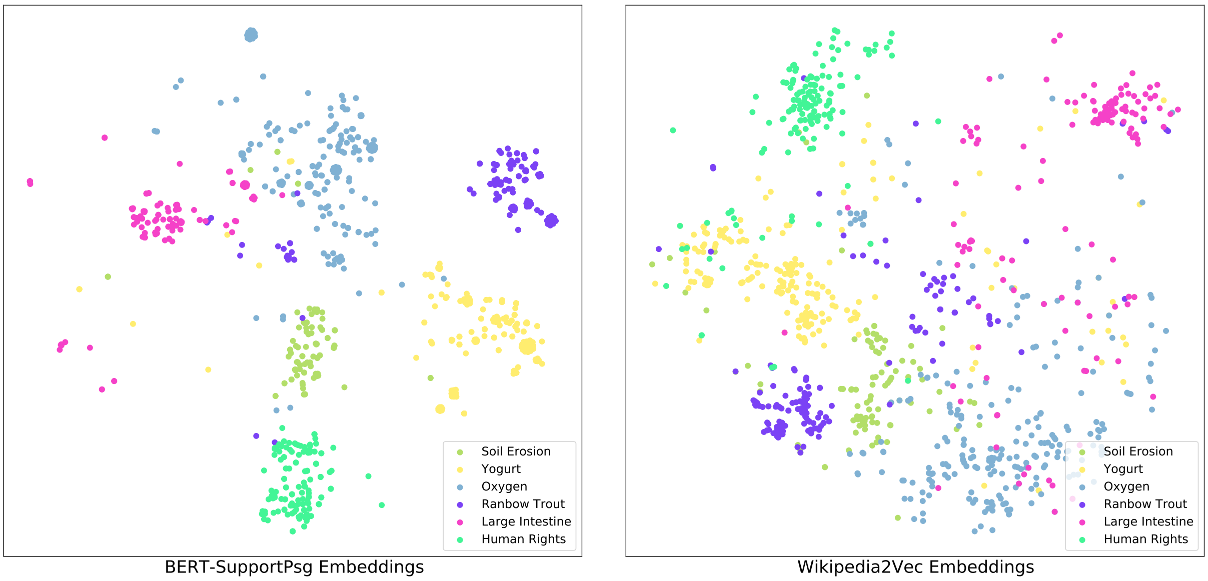


Figure 7.4: Visualizing clusters of relevant entities using t-SNE. We observe that relevant entities are better clustered using BERT-SupportPsg (left) than using Wikipedia2Vec (right).

Take-away. Regarding **RQ1**, it is not always sufficient to use the lead text of an entity as the entity's description; query-specific entity descriptions are better.

Regarding **RQ2**, although BERT-LeadText++ often performs well, our system BERT-ER++ using query-specific entity descriptions improves entity ranking performance by 13–42% over BERT-LeadText++. On its own (without other entity features), BERT-ER outperforms not only BERT-LeadText but also entity rankings obtained using Wikipedia2Vec, ERNIE, and E-BERT. This performance boost happens because our system can promote relevant entities to the top of the ranking while demoting non-relevant entities to the bottom. This happens because query-specific descriptions help our model to learn query-relevant information and minimize the non-relevant information.

7.5.3 BERT-ER Versus Wikipedia2Vec

Gerritse et al. [50] have shown that entity embeddings from a recent graph embedding approach called Wikipedia2Vec [155] are useful for entity ranking. As our work heavily relies on Wikipedia and uses Wikipedia as a Knowledge Base, we compare the performance of our query-specific BERT-ER to Wikipedia2Vec. We use the Wikipedia2Vec entity embeddings with the graph component made available by Gerritse et al. [50].[7].

Overall Results. From Table 7.4, we observe that BERT-ER outperforms Wikipedia2Vec on both CAR BenchmarkY1-Train and DBpedia-Entity v2. When performing the difficulty test described in Section 7.5.2, we find that BERT-ER obtains $MAP = 0.30$ for queries in the lower 0–50% range where Wikipedia2Vec obtains $MAP = 0.05$. Moreover, considering the query-specific descriptions individually, we observe that BERT-SupportPsg and BERT-Aspects consistently outperform Wikipedia2Vec on both datasets. This suggests that compared to Wikipedia2Vec, the entity embeddings obtained from BERT using query-specific entity descriptions capture the similarity/relevance of the entity for the query in a better way.

To verify this, we inspect the entity rankings for the query GMO obtained using Wikipedia2Vec and BERT-SupportPsg in Table 7.4. We find that BERT-SupportPsg places the relevant entity "Robert Swanson" at the top of the ranking (rank 3) compared to Wikipedia2Vec that places the entity at the bottom (rank 8). Moreover, BERT-SupportPsg demotes the non-relevant entity "Developmental Biology" that is placed higher by Wikipedia2Vec (rank 3) to the bottom of the ranking (rank 10). Our intuition is that this happens because entity embeddings obtained using BERT-SupportPsg are query-specific and encode the query-relevant knowledge about the entity that is helpful for determining the entity's relevance for the query. On the other hand, Wikipedia2Vec encodes the general knowledge about the entity available on Wikipedia, most of which is non-relevant in the context of the query.

Context-Dependent Entity Relatedness. As discussed in Section 7.1, queries and documents are often matched in the entity-space through the cosine similarity of the embeddings of entities mentioned in the query and document. Hence, it is important that the entity embeddings be able to capture context-dependent similarity between entities. For example, the entities "Food and Drug Administration" and "Robert Swanson" are related in the context of GMOs since Robert Swanson was the founder of the company that produced the first genetically engineered insulin approved for use by the Food and Drug Administra-

[7] Available from: `https://github.com/informagi/GEEER`

tion. We find that compared to Wikipedia2Vec, our query-specific BERT entity embeddings capture this context-dependent similarity between two entities in a better way. For example, compared to Wikipedia2Vec, BERT-SupportPsg assigns a higher similarity to the entities above.

Clustering Entities using Embeddings. As an additional evaluation, we assess whether the embeddings satisfy the cluster hypothesis [69]: documents (entities) relevant to a query cluster together. We consider the embeddings of relevant entities as points to be clustered and evaluate the quality of the resulting clusters. We use the following three metrics for evaluation: David-Bouldin score [27] (lower scores better), Silhouette score [119] (higher scores better), and Calinski-Harabasz score [16] (higher scores better).

From Table 7.5, we observe that clusters formed using embeddings from BERT-SupportPsg are better than clusters formed using Wikipedia2Vec. We also present a t-SNE [132] visualization of the resulting clusters for some example queries in CAR BenchmarkY1-Train. As we observe from Figure 7.4, the relevant entities for a query (e.g., "Yogurt", and "Oxygen") are close together, and the clusters are better separated using BERT-SupportPsg than using Wikipedia2Vec.

Take-away. Regarding **RQ3**, our query-specific BERT-ER outperforms Wikipedia2Vec on all datasets. BERT-ER finds relevant entities for the (difficult) queries for which Wikipedia2Vec fails because compared to Wikipedia2vec, BERT-ER captures the context-dependent similarity between query-entity pairs in a better way. BERT-ER can promote relevant entities to the top of the ranking while demoting the non-relevant entities to the bottom. This also ties back to RQ1 and RQ2: As BERT-ER uses query-specific entity descriptions, it enables BERT to focus on the query-relevant information to learn the embeddings of entities. As a result, these embeddings can differentiate between relevant and non-relevant entities better than Wikipedia2Vec.

Table 7.5: Results on BenchmarkY1-Train for clustering relevant entities. Evaluation measures: David-Bouldin score (lower better), Silhouette score (higher better), and Calinski-Harabasz score (higher better).

	David-Bouldin	Silhouette	Calinski-Harabasz
BERT-SupportPsg	3.87	-0.03	22.75
Wikipedia2Vec	5.29	-0.12	20.30

7.6 Conclusion

We present BERT-ER, query-specific BERT Entity Representations learnt by fine-tuning BERT for the entity ranking task. In contrast to the prevalent approach of using the static lead text from an entity's Wikipedia page as the entity's description, we study the utility of three types of *query-specific* entity descriptions: pseudo-relevant candidate passage, entity support passage and entity aspect.

Using BERT-ER for entity ranking, we obtain a performance improvement of 13–42% (MAP) over a system using the lead text as the entity's description, across a diverse range of queries from two large-scale entity ranking test collections. We also outperform entity ranking systems using Wikipedia2Vec, E-BERT, and ERNIE. We show that query-specific descriptions help an entity ranking system by promoting relevant entities to the top of the ranking, thereby increasing the precision at the top of the ranking. We also demonstrate that compared to Wikipedia2Vec, BERT-ER representations can identify when entities are related in the context of the query in a better way. We also show, both qualitatively and quantitatively, that compared to Wikipedia2Vec, our query-specific BERT-ER produce better clusters of relevant entities.

In the long-term, we believe that our approach to query-specific entity representations will lead to significant improvements in diverse IR and text analysis tasks, including question answering, and summarization. By demonstrating the importance of query-specific entity descriptions, we hope to promote more research in this area.

CHAPTER 8

ENTITY SALIENCE AND ENTITY RELATEDNESS FOR ENTITY ASPECT LINKING

8.1 Introduction

In Chapters 6 and 7, we leveraged entity aspects for obtaining fine-grained knowledge about entities from the context in which the entities have been mentioned. We showed that entity aspects and entity aspect links are useful for learning query-specific entity representations and for entity ranking. Although we briefly touched upon and introduced entity aspects and entity aspect linking in Chapters 6 and 7, in this chapter, we formally introduce them to the reader and explore the utility of entity salience and (static) entity relatedness measures for the task of entity aspect linking.

8.1.1 Motivation

Consider a journalist writing an article on the Coronavirus Disease 2019 (COVID-19) pandemic in which she analyzes the different angles of the pandemic on the economy and worker's safety. COVID-19 has been widely discussed in the news, social media, and research commentary in various aspects such as transmission, pathology, experimental treatment, food safety, protests, and stay-at-home-orders. So the journalist has to sift through hundreds of associated texts to identify those that discuss the right context. The journalist finds it challenging to identify different texts that are relevant for her article without being overwhelmed with other aspects of COVID-19. Of course, keyword searches help her, but she would also like to understand the larger connections between her topic and other policies, research findings, and incidents. One relevant text passage for her task is displayed in Figure 8.1, top.

Several **meat processing plants** around the **U.S.** are sitting idle this week because workers have been infected with the **coronavirus**. **Tyson Foods**, one of the country's biggest **meat processors**, says it suspended **operations** at its **pork plant** in **Columbus Junction, Iowa**, after more than two dozen workers got sick with **COVID-19**. **National Beef Packing** stopped **slaughtering cattle** at another **Iowa** plant, and **JBS USA** shut down work at a **beef plant** in **Pennsylvania**.

- Meat packing industry: Meatpackers / Today
- United States: Culture / Food
- Coronavirus Disease 2019: Cause / Transmission
- Tyson Foods: Controversies / Coronavirus (COVID-19) pandemic
- Continuous production: Shut-downs / Safety
- Columbus Junction, Iowa: History
- National Beef: Food Safety
- Cattle: Economy / Cattle meat production
- Iowa: Economy / Manufacturing
- Meat Industry: Effects on Livestock Workers
- Animal Slaughter: National laws / United States
- JBS USA: Coronavirus Outbreak
- Pennsylvania: Economy / Agriculture

Figure 8.1: Entity Aspect Linking Example. Top: An example paragraph about a COVID-19 outbreak in the meat packing industry, where the entity [Coronavirus Disease 2019] is mentioned with its aspect "transmission". Bottom: Entities mentioned in text with their referenced aspects (here taken from Wikipedia sections). From the fine-grained entity aspect links, the relevance of this paragraph for the journalist's task emerges. The paragraph is taken from NPR `https://n.pr/35XMdli`.

For such support systems, entity linking tools [41, 83, 104] identify and disambiguate entity mentions, such as of the entity "Coronavirus Disease 2019." However, the resulting entity links do not differentiate between different *aspects* of the entity that are being discussed. The journalist in the motivating example would be best helped with a fine-grained extension of entity links—we call this task entity aspect linking. Given a catalog of aspects for each entity, entity aspect linking predicts for each entity, which of its aspects is referenced in the context.

Figure 8.1 lists the most relevant entity aspects for the entities mentioned in the text above. In this example, the catalog of entity aspects is derived from sections in the Wikipedia article of the mentioned entity—other sources for aspect catalogs can be a reputation management platform or journalistic notes, as long as a brief description of each aspect is available. We refer to this description as *aspect content*.

For very popular topics such as COVID-19, aspect prediction can be addressed with lexicalized text classification. However, our goal is to develop a support system that also works for less popular topics, where the manual annotation of training data would defeat the purpose.

Finally, the benefits of an entity-aspect-linker go much beyond classifying into different aspects of a single target entity (such as COVID-19). While knowledge graphs have shown a lot of advantages, entity aspect linking allows to build a sub-entity knowledge graph, where nodes represent entity aspects, which are grouped into entities. Because the aspects are associated with content that mentions other entities, we can establish relations between entity-aspects by entity-aspect-linking the content of aspects.

Entity Aspect Linking Task. Given a mention M_E for entity E in a context C such a tweet, sentence or paragraph and a set of n predefined aspects $A_E = \{A_1, A_2, A_3, \cdots, A_n\}$ along with their contents, link the mention M_E to an aspect $A_i \in A_E$ that best captures the addressed topic.

In the following, we distinguish between context and content: the context is the text surrounding the entity mention we seek to aspect-link. With content, we refer to content

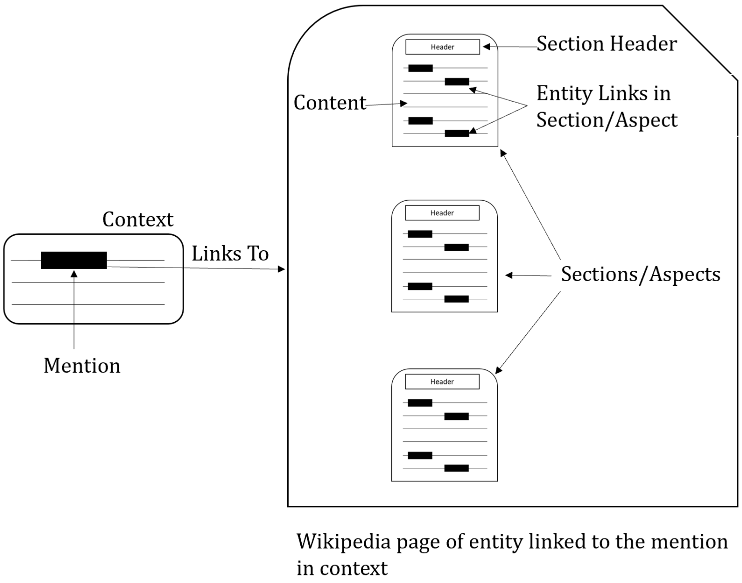

Section Header
Header
Entity Links in Section/Aspect
Content
Context
Links To
Sections/Aspects
Mention
Wikipedia page of entity linked to the mention in context

Figure 8.2: Graphical representation of various concepts in aspect linking.

associated with the aspect. This is depicted in Figure 8.2.

8.1.2 Research Gap

Nanni et al. [91] introduce the entity aspect linking task and suggest a combination of text similarity metrics between mention context and aspect content. One of the similarity metrics includes the overlap and similarity of other entities mentioned in context and aspect content. However, their approach would consider all entities equally important for the aspect-linking decision. We believe the approach can be improved by incorporating the entity salience and entity relatedness.

In context and content, only few entities are salient (that is, central), while most other entities are mentioned in passing, such as in examples, circumstantial references, or clarifications. We alleviate this problem by incorporating the salience of the entity in the aspect content. We hypothesize that incorporating the salience of the entity while aspect linking would improve performance. Many approaches for entity salience detection have been developed recently [37, 109, 150]. In this work, we use the state-of-the-art salience detection system from Ponza et al. [109] to incorporate the salience of the entity in aspect linking. We present several entity salience-based indicators and show that it is indeed useful to incorporate the salience of an entity along with the lexical and semantic indicators in entity aspect linking. We present a detailed analysis of the conditions under which salience works and fails.

In context and content, many entities are closely related but will not be exact matches. We alleviate this by incorporating the relatedness of the entities in the aspect content to the target entity (the entity we are trying to aspect link). We hypothesize that aspects with content which have many highly related entities to the target entity are more likely expressions of the right aspect. We explore if entity relatedness is helpful to link contexts to aspect content whenever they are not mentioning identical entities, but entities that are highly related. Several entity relatedness measures have been suggested which predict the similarity between two entities [108, 115, 120, 158]. We experiment with using a state-of-the-art entity relatedness detection system from Piccinno et al. [104]. We show that using a measure of entity relatedness from an external tool to match entity aspects, when used in combination with the lexical and semantic features can improve performance on several baselines.

8.1.3 Contributions

Our contributions are as follows:

1. We present a novel method of EAL which incorporates the salience of the target entity in the aspect. We show that entity salience detection can help by learning information which is complementary to some lexical and semantic features and provides an improvement of 10% on the existing state-of-the-art for EAL.

2. We show that entity relatedness based indicators are not only strong on their own but also help to boost retrieval performance on the EAL task by improving retrieval effectiveness by upto 8% when used in conjunction with entity salience, lexical and semantic indicators.

3. We show that although a static measure of relatedness from an off-the-shelf tool can help the task, the performance is inferior to using the frequency of co-occurrence of an entity with another entity as a relatedness measure.

8.1.4 Outline

The remainder of this chapter is organized as follows. As our method builds on the features suggested by Nanni et al. [91], we briefly describe them in Section 8.2. In Section 8.3, we describe our approach to entity aspect linking using entity salience and entity relatedness. In Section 8.4 we describe the experimental methodology, followed by a discussion of the results in Section 8.5. We end the chapter with Section 8.6.

8.2 Background: EAL Method of Nanni et al.

Nanni et al. [91] uses a "bag of words" vector space model to represent entity aspects. They consider three different ways of comparing the context of the entity mention based on the following entity aspect fields:

1. **Header.** Rank aspects based on similarity of the mention in context to the header of each section in the Wikipedia page.

2. **Content.** Rank aspects based on the similarity between the mention in context and the content of each section of the Wikipedia page of the entity.

3. **Entity.** Overlap of entities mentioned in the context of the entity mention and the content of a section on the Wikipedia page of the entity.

They use five methods to derive features from these fields:

1. **TF-IDF.** Cosine similarity between the TF-IDF (logarithmic, L2-normalized) vector of contextual mention and aspect field.

2. **BM25.** Rank aspect representations using the contextual mention as a query using BM25 ($k_1 = 2, b = 0.75$).

3. **Word Embeddings.** Cosine similarity between the mention in context and the aspect using pre-trained GloVe [101] embeddings of dimension 300.

Passage 1. The British government came under heightened pressure to disclose details about a secretive scientific advisory group after a report on Friday that a top political aide to Prime Minister Boris Johnson had taken part in the group's meetings on the coronavirus pandemic.
Passage 2. British Prime Minister Boris Johnson is resisting growing calls to reopen the UK from its lockdown because he is still so "frightened" from his own near-fatal brush with the bug, according to a report .

Figure 8.3: Salient versus Non-Salient Passage. We notice that Passage 2 discusses how the entity *Boris Johnson* in his role as the Prime Minister of the UK is affecting the pandemic situation whereas Passage 1 just mentions the entity on the side. We say that *Boris Johnson* is *salient*, i.e., central in Passage 2.

4. **Entity Embeddings.** Using 500 dimensional RDF2Vec [115] embeddings to embed entities in the context of the entity mention and a section from the Wikipedia page of the entity, then compute the document vector using the TF-IDF of an an entity in context of the entity mention and its embedding.

5. **Term Overlap.** Number of shared words after tokenization.

6. **Size.** a context-independent feature, which is the numbers of words in an entity aspect's text.

Furthermore, they explore different context sizes: sentence, paragraph, and section.

8.3 Approach

Our goal is to enrich entity links with fine-grained aspects of the entities. In this work, we incorporate entity importance through entity salience and relatedness. We refer to the entity that we want to link to the correct aspect link as the *target entity*. In the following, we discuss several ideas as micro-approaches. In the evaluation, we will compare them on their own, and through a supervised combination that uses each as a feature.

8.3.1 Entity Salience for Entity Aspect Linking

Our first extension is to incorporate indicators from entity salience detection. As discussed in Section 8.1.2, only few entities are salient in the content and context while most other

Aspect. Prime Minister of UK
Content. Former Prime Minister Theresa May has criticised world leaders for failing "to forge a coherent international response" to the coronavirus pandemic. Mrs May's intervention comes as Boris Johnson and Sir Keir Starmer face each other at Prime Minister's Questions for the first time later.

Figure 8.4: Example aspect with content to depict entity relatedness intuition. This aspect may be a good aspect for the entity *Boris Johnson* since it mentions several related entities such as *Theresa May* and *Sir Keir Starmer*.

entities are mentioned as a passing reference (see Figure 8.3). Hence, we would prefer to link an entity to the aspect in which the entity is salient. We hypothesize that entity salience is a useful indicator of aspects for entities. Due to its superior performance and ease of use, we use the entity salience detection system from Ponza et al. called SWAT [109] to find the salience of an entity in text. Given some text, SWAT returns the entities along with their salience scores in the text. We use salience detection in our aspect-linking approach by incorporating the following indicators.

1. **Salience of Entity Mention in aspect (Sal-EM).** Score of the aspect is equal to the salience score of the entity mention in the aspect if the entity is salient, and zero otherwise.

2. **Salient Entities in Context (SEC).** Score an aspect by summing over the salience score of entities $e \in E$, where $E = E_A \cap E_C$, $E_A =$ aspect entities, and $E_C =$ salient entities in sentence, paragraph and section context.

3. **All Entities in Context (AEC).** To investigate the extent to which salient entities can affect the performance, we also experiment with using both salient and non-salient entities from SWAT for E_C in (2) above.

8.3.2 Entity Relatedness for Entity Aspect Linking

Entity Relatedness is a measure of how strongly related two entities are. For example, consider the entities, *Boris Johnson*, *Theresa May*, and *Donald Trump*. Intuitively, one would say that *Boris Johnson* is more strongly related to *Theresa May* than to *Donald Trump* because both *Boris Johnson* and *Theresa May* are British politicians. We hypothesize that this measure of entity relatedness can help in aspect linking. More concretely, we hypothesize that an aspect mentioning many related entities to the target entity is a good candidate for an aspect for the given target entity (see Figure 8.4.

We use two measures of relatedness in this work:

1. **Frequency of co-occurring entities.** We assume that entities that co-occur frequently with the target entity are highly related to the target entity. We calculate the relatedness of an entity e_2 to an entity e_1 by counting the number of times e_2 is mentioned in the context of e_1.

2. **Relatedness from an external tool.** Due to its superior performance and ease of use, we use the Entity Relatedness system from Piccinno et al. called WAT [104] to find relatedness between pairs of entities. Given a list of entities, WAT provides the relatedness measure between every pair of entities in the list.

Co-occurring entities with an entity from context. The current state-of-the-art method for EAL from Nanni et al. [91] uses exact matches of entities in the context and aspect content. However, since there are only few such entities, we explore methods to derive an expanded set of prominent entities e', then rank aspects by how frequently they match these most prominent entities e'.

For every entity e in the context of the target entity e_T (sentence, paragraph or section), we retrieve top-k[1] passages from a background corpus using e's name as a query. We construct a bag of potential expansion entities (henceforth called PROM in this paper) from all entities mentioned in all $E \times k$ passages that were retrieved through E queries, one for each entity e in the context. We calculate the frequency $\text{tf}_e(e')$ with which entities occur in the passage ranking for entity e as follows:

$$\text{tf}_e(e') = \sum_{p \in P_e} \text{tf}(e'|p) \tag{8.1}$$

where P_e are the passages mentioning e in the passage ranking obtained using e as query and $\text{tf}(e'|p)$ is the frequency of e' in p. From this frequency, we derive a prominence $P(e'|e_T)$

[1] We use $k = 100$.

and use it to reweigh entities in this bag. The prominence not only incorporates the degree of association of e' with the contextual entity e, but also how consistently e' is associated with other entities e that occur in the context we seek to link. Akin to language models, the term frequency $\text{tf}_e(e')$ is normalized to sum to one over all context entities e''.

$$P(e'|e_T) = \frac{1}{E} \sum_{e} \sum_{e'} \frac{\text{tf}_e(e')}{\sum_{e''} \text{tf}_e(e'')} \tag{8.2}$$

Finally, we match prominent entities e' in target aspects a: the more frequent prominent entities e' are mentioned in the aspect content, the higher the aspect is ranked.

$$\text{Score(a)} = \sum_{e' \in a} P(e'|e_T) \tag{8.3}$$

We explore three variants of this approach.

1. **Simple Frequency Distribution (SF-Dist).** Using the prominence score $P(e'|e_T)$ produced via passage ranking of all entities e mentioned in the context. This co-occurrence-based entity scoring metric allows us to match more entities e' in the aspect than solely using entities in the context e. However, as any expansion approach, it is prone to promote spurious matches.

2. **Weighted Frequency Distribution (WF-Dist).** To remedy the promotion of spurious matches, we combine the prominence score with an entity-relatedness score between entities e' and the target entity e_T.

$$\text{Score(a)} = \sum_{e' \in a} \text{Relatedness}(e', e_T) \cdot P(e'|e_T) \tag{8.4}$$

3. **Relatedness Distribution (Rel-Dist).** Same approach as in Equation 8.4 but the prominence score is ignored. We only use the prominence approach to select a set of entities e'.

Co-occurring entities with the target entity. Since contextual entities e might be less well suited than the target entity e_T, we explore several methods that emphasize a connection to the target entity.

1. **Rel-Dist-PROM.** Using the prominence approach above, but building an entity set only for the target entity e_T, while ignoring other contextual entities e.

2. **Rel-Dist-Wiki.** Instead of identifying entities through other passages, we consider entities e' mentioned on the Wikipedia page of the target entity e_T. These entities e' are ranked by relatedness to e_T.

Instead of ranking aspects a by a sum of prominence/relatedness scores that they contain, we invert the approach and rank aspects by retrieval models. The query is formed by the title of the target entity which is expanded with top 20 entities e' — these are weighted in the query akin to RM3 [70]. We explore weighted combinations of the following retrieval models as implemented in Lucene: (1) BM25 (default parameters), (2) Language Models with Jelinek-Mercer (LMJM) smoothing $(\lambda = 0.4)$, and (3) Language Models with Dirichlet (LMDS) smoothing, with both RM1 and RM3 expansion models on entities.

1. **RS-Asp-Freq-PROM.** Expand query weighted by prominence score and retrieve aspects.

2. **RS-Asp-Rel-PROM.** Expand query weighted by relatedness score and retrieve aspects.

3. **RS-Asp-Rel-Wiki.** Expand with top 20 entities by relatedness (to e_T) on Wikipedia page of e_T and retrieve aspects.

8.4 Evaluation

In this section, a quantitative evaluation of our system for the Entity Aspect Linking (EAL) task is presented on the entity-aspect dataset from Nanni et al. [91]. We begin by giving a brief overview of the datasets used in this work in Section 8.4.1. We then describe our experimental settings (Section 8.4.2) followed by the baselines (Section 8.4.3). We end the section by presenting some research questions pertaining to three broad components of our system: Entity Salience, Entity Relatedness and Co-occurring entities, which our experiments aim to address (Section 8.4.4).

8.4.1 Entity Aspect Linking Benchmark

We use the Entity Aspect Linking dataset from Nanni et al. [91]. It consists of 201 entity mentions from Wikipedia along with their sentence, paragraph and section context, and a list of candidate aspects for the mention. We use this dataset for training a Learning-to-rank (L2R) algorithm using 5-fold cross validation.

Nanni derived this dataset from hyperlinks on Wikipedia pages that refer to sections on other pages after omitting hyperlinks to administrative sections such as “References”, “See Also”, etc. Each training example contains the sentence context, paragraph context, entity mention and candidate aspects. The candidate aspects are the top-level sections of the Wikipedia page of the entity linked to the mention. Each candidate aspect contains information about its section header, text, and entities that are contained within it. In addition to the entity links in both context and aspect content which are provided with the dataset, we use WAT[2] [104] to annotate additional entity links in the context and aspect content.

Passage Corpus. We use the corpus of paragraphs from the TREC Complex Answer Retrieval (CAR) track [34][3] dataset as a source of passages when determining the prominence score in Section 8.3.2. It consists of an entity linked corpus consisting of paragraphs from the entire English Wikipedia. We construct a Lucene index of passages and use the top 100 passages retrieved with BM25 (Lucene default) using the entity name as the query.

Ground Truth. The dataset from Nanni et al. [91] contains contexts, entity links to targets, and the correct aspect. The dataset was automatically created and manually verified.

[2] WAT has both, an entity linking system and an entity relatedness prediction system. See `https://sobigdata.d4science.org/web/tagme/wat-api`

[3] `http://trec-car.cs.unh.edu`

8.4.2 Evaluation Paradigm

Evaluation Metrics. Our methods predict a ranking of aspects. We use Precision at 1 (P@1) and Mean Reciprocal Rank (MRR) as our evaluation metrics.

Machine Learning. We apply our methods to produce an aspect ranking for every entity mention. We then treat each ranking as a feature and perform 5-fold cross validation with a list-wise learning-to-rank (L2R) method (Coordinate Ascent) optimized for Precision at 1 (P@1). We use RankLib[4] for this purpose.

Feature Subsets We present an ablation study with various subsets of features. Below, we define the feature subsets we use in our study.

- **Subset-1.** All features based on entity relatedness.
- **Subset-2.** All features based on entity salience.
- **Subset-3.** All features based on entity relatedness along with lexical and semantic features from Nanni et al.
- **Subset-4.** All features based on entity salience along with lexical and semantic features from Nanni et al.
- **Subset-5.** All features based on entity relatedness and entity salience along with lexical and semantic features from Nanni et al.

8.4.3 Baselines

We include the following baselines in this work:

1. **Nanni's method.** We re-implement all lexical and semantic features from Nanni et al. [91] and use a supervised combination of sentence, paragraph and section context features as our baselines. Their method is discussed briefly in Section 8.2.

[4] http://sourceforge.net/p/lemur/wiki/RankLib

2. **Size.** We consider the length of each section (in number of tokens) and link the entity-mention to the longest.

8.4.4 Research Questions

We study the following research questions in this chapter:

RQ1 To what extent can entity salience help EAL?

RQ2 To what extent can entity relatedness help EAL?

RQ3 Is the frequency or relatedness of co-occurring entities a better indicator of aspects?

8.5 Results and Discussions

The results from our experiments are presented in Table 8.1. Below, we discuss each research question presented in Section 8.4.4. Note that in our work, we treat frequency of co-occurrence as a measure of relatedness between two entities, with the assumption that more frequent entities are also highly related. In addition, we also use the relatedness measure obtained from an external tool [104] and compare which contributes more to the task. In the discussion that follows, we use the term *relatedness distribution* to denote the relatedness distribution over co-occurring entities which is obtained using the external tool, and the term *frequency distribution* to denote the relatedness distribution obtained by counting how frequently an entity co-occurs with the target entity.

8.5.1 Entity Salience

From Table 8.1, we observe that a supervised combination of all salience features (Subset-2, P@1 $= 0.59$) outperforms two of the four baselines, *Nanni et al. (Section)* (P@1 $= 0.53$) and *Size* (P@1 $= 0.39$). We also observe that a combination of all salience features with the lexical and semantic features (Subset-4, P@1 $= 0.72$) outperforms all baselines and provides an improvement of 10% over the best performing baseline, *Nanni et al. (Sentence)* (P@1 $= 0.65$). However, considering all entities (salient and non-salient) in the context

Table 8.1: Performance with standard error of individual entity salience and relatedness features and combined with L2R, including subsets/ablations.

Method	P@1	MRR	Success@5	NDCG@10
Nanni et al. (Sentence)	0.67±0.03	0.79±0.02	0.97±0.03	0.85±0.03
Nanni et al. (Paragraph)	0.64±0.03	0.78±0.03	0.99±0.03	0.84±0.03
Nanni et al. (Section)	0.53±0.03	0.71±0.03	0.97±0.03	0.78±0.03
Size	0.39±0.03	0.60±0.03	0.93±0.03	0.70±0.03
Sal-EM	0.19±0.03	0.46±0.03	0.82±0.03	0.58±0.03
SEC (Sentence)	0.23±0.03	0.53±0.03	0.85±0.03	0.63±0.03
AEC (Sentence)	0.51±0.03	0.70±0.03	0.95±0.03	0.77±0.03
SF-Dist (Sentence)	0.54±0.03	0.72±0.03	0.96±0.03	0.79±0.03
WF-Dist (Sentence)	0.49±0.03	0.67±0.03	0.94±0.03	0.75±0.03
Rel-Dist (Sentence)	0.43±0.03	0.62±0.03	0.94±0.03	0.72±0.03
SF-Dist-PROM	0.35±0.03	0.59±0.03	0.93±0.03	0.69±0.03
Rel-Dist-PROM	0.36±0.03	0.60±0.03	0.90±0.03	0.69±0.03
RS-Asp-Freq-PROM (LMJM + RM1)	0.35±0.03	0.59±0.03	0.90±0.03	0.67±0.03
RS-Asp-Rel-PROM (LMJM + RM1)	0.40±0.03	0.60±0.03	0.91±0.03	0.69±0.03
Rel-Dist-Wiki	0.37±0.03	0.60±0.03	0.92±0.03	0.69±0.03
RS-Asp-Rel-Wiki (BM25 + RM3)	0.39±0.03	0.61±0.03	0.93±0.03	0.70±0.03
Subset-1 (Only Relatedness)	0.48±0.03	0.68±0.03	0.96±0.03	0.75±0.03
Subset-2 (Only Salience)	0.59±0.03	0.72±0.03	0.93±0.03	0.78±0.03
Subset-3 (Rel. + Lex. + Sem.)	0.66±0.03	0.78±0.03	0.97±0.03	0.83±0.03
Subset-4 (Sal + Lex. + Sem.)	0.72±0.03	0.82±0.03	0.97±0.03	0.87±0.03
Subset-5 (Sal. + Rel. + Lex. + Sem.)	0.70±0.03	0.81±0.03	0.97±0.03	0.85±0.03

(AEC Sentence, P@1 $= 0.51$) performs better than considering only the salient entities (SEC Sentence, P@1 $= 0.23$).

Salient versus non-salient entities. These observations show the effectiveness of using salience. However, they also indicate that considering non-salient entities together with the salient ones help improve performance. To investigate this further, we manually confirmed that SWAT correctly identifies salient entities in text. However, SWAT returns many empty results when asked for only salient entities than when asked for all (salient or otherwise) entities. For example, using the sentence context of an entity mention, SWAT returns an empty result for 100 of the 201 entity mentions when asked for only the salient entities and 13 of 201 entity mentions when asked for all entities. This shows the limitations of SWAT and why the results obtained using *SEC (Sentence)* is lower than *AEC (Sentence)*.

Our intuition is that the other entities, although non-salient, have some inherent semantic meaning and hence considering them together with the salient entities helps the task. This is the case for the paragraph and section contexts too but we do not show the results here due to space constraints.

Issue with exact entity matching. One issue with matching entities in such an exact way is that very often, there are no matches. In such cases, a lot of aspects receive a score of zero in our methods (SEC and AEC in Section 8.3.1). This also shows why the *AEC (Sentence)* outperforms *SEC (Sentence)* in Table 8.1. There are more matching entities when matching all aspect entities (as in AEC) than when matching only salient aspect entities (as in SEC), with salient context entities. To illustrate this issue, we present an example from our dataset in Figure 8.5. We observe that the entity *Kyoto Protocol* is salient in the context (shown by bold italic) but not in the content (shown by only bold). Hence, SEC would score this aspect zero since there are no matching salient entities, but AEC would not.

Difficulty test. To investigate the extent to which salience helps, we divide the entity mentions into different levels of difficulty according to the performance (P@1) of the *Nanni et al. (Sentence)* method, with the 5% most difficult queries for this method to the left and the 5% easiest ones to the right, and compare the performance with the *Subset-4*. The results are shown in Figure 8.6.

Take-Away. With respect to **RQ1**, we can say that entity salience does indeed affect the task positively. We are able to outperform all the baselines with the help of salience and achieve an improvement of 10% on the best performing baseline. We see that salience helps to boost performance when the aspect linking decisions get difficult by learning information which is complementary to the lexical and semantic features. However, exactly matching entities between context and aspect content leads to finding no matches for most aspects and hence most aspects receive a score of zero. Added to this are the limitations of SWAT in finding salient entities. Hence, entity salience is helpful but needs to be balanced

Mention. Annex 1
Sentence Context. Under the ***Kyoto Protocol***, the ’caps’ or quotas for Greenhouse gases for the developed Annex 1 countries are known as ”’Assigned Amounts”’ and are listed in Annex B.
Aspect Content. ***The United Nations Climate Change Conference*** are ***yearly conferences*** held in **the framework** of the **UNFCC**. They serve as ***the formal meeting*** of **the UNFCC** Parties (”**Conferences** of **the Parties**”) (**COP**) to assess **progress** in dealing with **climate change**, and beginning in **the mid-1990s**, to negotiate **the Kyoto Protocol** to establish legally binding **obligations** for developed **countries** to reduce their **greenhouse gas emissions**.

Figure 8.5: Example mention with SWAT annotated sentence context and aspect content. The salient entities are in bold italic and the non-salient ones are in bold.

with additional entities to be effective.

8.5.2 Entity Relatedness

We observe the following from Table 8.1.

1. A supervised combination of all relatedness features (Subset-1, P@1 $= 0.48$) does not perform very well on its own, doing better than only one baseline, *Size* (P@1 $= 0.39$).

2. A combination of relatedness features with the lexical and semantic features (Subset-3, P@1 $= 0.66$) does significantly better than all baselines. However, the improvement over the best performing baseline (*Nanni et al. (Sentence)*, P@1 $= 0.65$) is little.

3. We achieve an improvement of 8% on the best performing baseline (*Nanni et al. (Sentence)*, P@1 $= 0.65$) by using a combination of salience and relatedness features with the lexical and semantic features (*Subset-5*, P@1 $= 0.70$).

4. Using only salience based indicators (Subset-2, P@1 $= 0.59$) works better than using only relatedness based indicators (Subset-1, P@1 $= 0.48$).

5. Adding relatedness based indicators to the mix results in a slight decrease in performance from P@1 $= 0.72$ on *Subset-4* to P@1 $= 0.70$ on *Subset-5*.

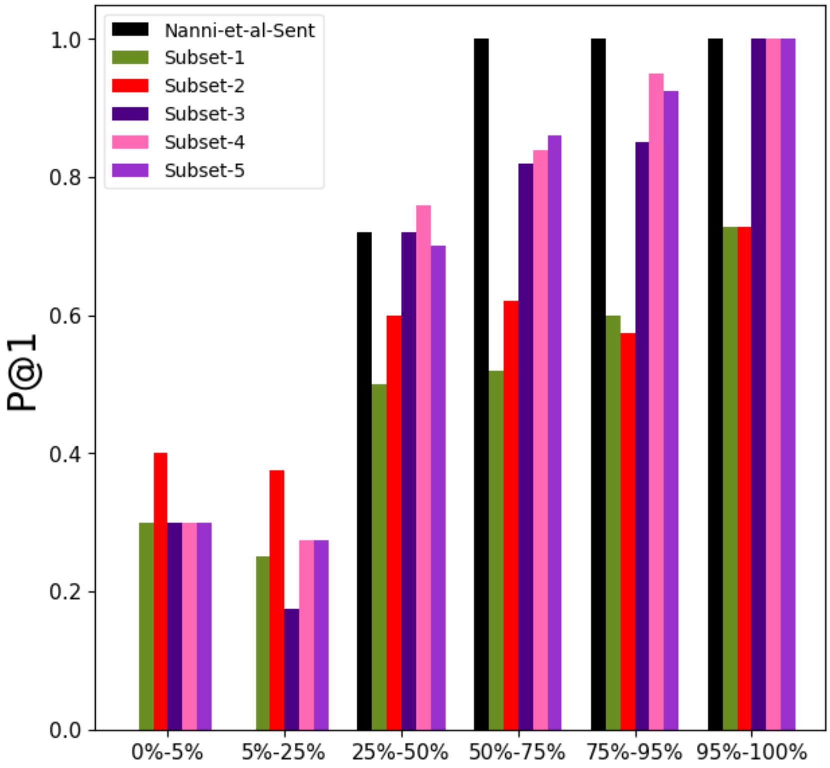


Figure 8.6: Difficulty-test for P@1, comparing Nanni et al.(Sentence) to various L2R systems. We observe that whenever it is difficult to perform the task using *Nanni et al. (Sentence)*, entity salience supports our L2R system (Subset-2 and Subset-4).

Drawbacks of WAT. These observations indicate that entity relatedness does indeed affect the task positively. However, salience is more informative than relatedness as is evident from the superior performance of Subset-2 over Subset-1 and Subset-4 over Subset-3. On further investigation, we found that WAT finds many false positives and false negatives. For example, given the entity list consisting of *World War I*, *Vietnam War* and *France*, it predicts that *World War I* is related to *Vietnam War* (false positive) but unrelated to *France* (false negative). This is because WAT does not take the query or the context of the entity into account but makes predictions based on graph-based features such as number of inlinks and outlinks to and from a particular entity node in a knowledge graph.

Difficulty Test. To investigate the extent to which relatedness helps, we present results from the difficulty test explained in Section 8.5.1 in Figure 8.6. We observe that whenever it is difficult to perform the task using *Nanni et al. (Sentence)*, entity relatedness supports our L2R system (Subset-3 and Subset-5).

Take-Away. With respect to **RQ2**, we may say that relatedness does indeed affect the task positively. Although the relatedness of entities by itself may not perform very well, it shows its strength in a supervised combination with lexical and semantic features leading to performance improvement of 8% over the best baseline. Moreover, a L2R system containing relatedness based features help to boost performance when aspect linking decisions get difficult. However, the limitations of WAT hinder the performance of a system using it.

8.5.3 Frequency vs Relatedness

We observe from Table 8.1 that ranking aspects using *SF-Dist (Sentence)* (P@1 $= 0.54$) outperforms *WF-Dist (Sentence)* (P@1 $= 0.49$), which in turn outperforms *Rel-Dist (Sentence)* (P@1 $= 0.43$).

Quality of entity rankings obtained using frequency and relatedness. These observations indicate that using the frequency of co-occurring entities as a relatedness measure is more informative than the relatedness obtained from WAT. To investigate this further and to answer RQ3, we produce entity rankings for every aspect using: (1) Frequency (SF-Dist in Section 8.3.2) and (2) Relatedness (Rel-Dist in Section 8.3.2) of entities. We evaluate these entity rankings using a “ground truth” of aspects where we define any entity mentioned in the aspect as relevant for the aspect. We use Mean-Average Precision (MAP) as our evaluation metric for this experiment.

We found that the entity rankings obtained using frequency distribution (prominence) are indeed better than those obtained using relatedness distribution. For example, entity ranking obtained using section context and frequency distribution has MAP $= 0.13$ whereas that obtained using relatedness distribution has MAP $= 0.04$.

Helps-hurts analysis. We also perform an additional experiment where we analyse the number of entities that were helped (in terms of P@1) by using frequency distribution of co-occurring entities as compared to using the relatedness distribution. We found that using frequency distribution over the co-occurring entities helps more queries than using a relatedness distribution. For example, with section, using frequency distribution helps 1160 aspects while hurting just 3 as compared to using relatedness. This shows why the entity ranking obtained using frequency and section context performs better than that obtained using relatedness, and consequently, why the *SF-Dist* methods perform better than the *Rel-Dist* method in Table 8.1.

Take-Away. With respect to **RQ3**, although relatedness from WAT can help when the aspect linking decisions get more difficult, as compared to frequency, it hurts the performance of a method using it.

8.6 Conclusion

This chapter addresses the task of entity aspect linking and studies the effectiveness of using entity salience and relatedness on the task using two off-the-shelf tools not trained for this task. We show that although these tools are not perfect and do not pose a solution on their own, a supervised combination of salience and relatedness features with lexical and semantic features can outperform several established baselines. In particular, we show that such a supervised combination learns complementary information which aids the performance of the supervised system. Moreover, we find that using the frequency of co-occurring entities as a relatedness measure between two entities is better than using their relatedness from an off-the-shelf-tool.

Despite this success, we believe that there is potential for further improvement, if salience detection and entity relatedness would be customized for the entity aspect linking task. One issue is that relatedness is both unaware of the context and the aspect content. Extending entity relatedness measures to consider relatedness-in-context (similar to the prominence score) is likely to offer further improvements. Analogously, salience detection is currently trained for a linguistic purpose, that is unaware of the downstream task. We speculate that developing a new salience-like component that can identify which entities in the context are sufficiently central to be incorporated in the matching decision. Both a context-ware entity relatedness and a task-aware salience detector—once available—

would also be useful for other downstream tasks.

PART III

CONCLUSION

This page intentionally left blank.

CHAPTER 9

SUMMARY

Through this dissertation, we advance the state-of-the-art in IR by developing neural entity-oriented search algorithms that understand text at a deeper level by obtaining a deeper understanding of entities in the text. We explore the utility of methods such as entity salience and (query-specific) entity relatedness whose utility for IR tasks were not well-studied previously, thus adding new knowledge to the field regarding which ideas/concepts work the best. With the development of deep learning for IR/NLP, emphasis is often placed on developing more complex models; however, thorough this dissertation, we find that (simple) retrieval-based approaches can yield significant performance improvements without increasing model/training complexity.

We explore two major research directions in this dissertation that inform each-other and together help to advance the state-of-the-art in IR: (1) Obtaining fine-grained and query-specific knowledge about entities from the context of entity mentions, and (2) Using this knowledge to automatically learn query-specific vector representations of entities. We obtain this query-specific knowledge from two sources: (1) entity-support passages retrieved from a text corpus, and (2) entity aspects (obtained using entity aspect linking) which are top-level sections from Wikipedia. We show that the two sources provide complimentary information about entities. Together, they allow us to understand the query-relevant aspects of an entity, which we leverage for text matching during retrieval (Chapters 6 and 7).

With respect to entity-support passage retrieval (Chapter 5), we explore the utility of entity salience for the task and show that salience is a high precision indicator that leads to improvements when applicable. As noted earlier, current entity-oriented search systems often use the introductory paragraph from Wikipedia as the entity's description. Through our work on entity-support passage retrieval, we show that significant performance improvements can be obtained when replacing (generic) information from Wikipedia with query-specific knowledge obtained through Pseudo-Relevance Feedback. Our intuition is that the Wikipedia page contains a lot of topics/information about the target entity but only some of this is relevant in the context of the given query. Hence, identifying and distinguishing this query-relevant information from the other information on the Wikipedia page leads to improvements.

Our research studies the question: How can we identify the query-relevant parts from Wikipedia? To this end, we identify the top-level sections (entity aspects) from the Wikipedia page of an entity that provides information about the entity in the context of the query. We leverage entity aspects for IR and evaluate their efficacy in the context of the entity ranking task in two ways: (1) Using entity aspects directly by deriving entity relevance features from them in Chapter 6, and (2) Using the entity aspects to learn query-specific entity representations using BERT in Chapter 7. In both cases, we find that using entity aspects leads to significant performance improvements over several state-of-the-art entity ranking systems (both neural as well as non-neural) on two large-scale entity ranking test collections.

Our query-specific entity representations are able to identify when two entities are similar in the context of the query even when they are apparently unrelated in the Knowledge Graph. We show this through our work on entity ranking in Chapter 7. Through this dissertation, we make a significant contribution to the emerging and growing field of learning BERT-based entity representations.

Entity ranking is a very important task in information retrieval: Often, queries such as *Where was Mother Teresa born?* can be answered using entities. Our work on entity ranking using the question-answering queries from DBpedia-Entity v2 dataset significantly advances the state-of-the-art in the area. In particular, we show that incorporating fine-grained query-specific knowledge into the entity retrieval system often leads to significant performance improvements. Further, we propose a new end-to-end entity ranking system that does not require extensive feature engineering and learns entity relevance indicators automatically from training data.

CHAPTER 10

FUTURE WORK

The research presented in this dissertation serves as a stepping-stone for further research in the field of neural entity-oriented information retrieval and extraction. The ideas presented in this dissertation has the potential to not only impact the field of IR but also other related fields such as Natural Language Processing and Question-Answering. Below, we discuss some possible future directions that one might pursue:

- **Query-specific entity representations for question-answering systems.** In this dissertation, we show how to obtain query-specific knowledge about an entity and use this knowledge to learn vector representation of the entity. Our work in learning query-specific entity representations has the potential to have significant impact on question-answering. Often, the first step in question-answering is Answer Sentence Selection: Selecting the sentence containing (or constituting) the answer to a question from a set of retrieved relevant documents. In such cases, the question and candidate sentence may be matched in the entity-space using our query-specific entity representations. Our intuition is that query-specific entity representations can improve on this component, hence improving the overall quality of the question-answering system.

- **Automatic hyperlinking of news articles.** In addition to providing links to articles that give the reader background or contextual information, journalists sometimes link mentions of concepts, artifacts, entities, etc. to internal or external pages with in-depth information that will help the reader better understand the article. Our work on entity-support passage retrieval can be used to automatically hyperlink entities, concepts, or references in news articles to another resource that provides more information on the linked thing.

- **Automatic clustering of news articles.** In the news domain, it is often required to cluster (long) news articles on the same topic together. This clustering method would need a measure of text similarity to perform well. Our research on entity aspect linking has the potential to impact this domain: Our research on entity aspect linking focuses on developing models that can match an aspect to the context, thus trying to model the similarity between the aspect and the context. This can aid the news clustering system.

Finally, we envision this dissertation to serve as a stepping stone towards building more intelligent information finding systems. Such systems would one day respond to a user's open-ended and complex information needs with a complete answer instead of a ranked list of results, thus (finally) transforming the "search" engine into an "answering" engine.

LIST OF REFERENCES

[1] Nitish Aggarwal, Sumit Bhatia, and Vinith Misra. Connecting the dots: Explaining relationships between unconnected entities in a knowledge graph. In *The Semantic Web. European Semantic Web Conference*, Lecture Notes in Computer Science, pages 35–39. Springer, Cham, 2016.

[2] Jay Alammar. The illustrated transformer. `https://jalammar.github.io/illustrated-transformer/`, June 2018.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. 2014.

[4] Krisztian Balog. *Entity-oriented search*, volume 39 of *The Information Retrieval Series*. Springer, 1 edition, 2018.

[5] Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 43–50, New York, NY, USA, 2006. Association for Computing Machinery.

[6] Krisztian Balog, Marc Bron, and Maarten De Rijke. Query modeling for entity search based on terms, categories, and examples. *ACM Trans. Inf. Syst.*, 29(4), December 2011.

[7] Krisztian Balog, Pavel Serdyukov, and Arjen P De Vries. Overview of the trec 2010 entity track. Technical report, Norwegian University of Science and Technology, 2010.

[8] Siddhartha Banerjee and Prasenjit Mitra. WikiKreator: Improving Wikipedia stubs automatically. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 867–877, Beijing, China, July 2015. Association for Computational Linguistics.

[9] Michael Bendersky, Donald Metzler, and W. Bruce Croft. Learning concept importance using a weighted dependence model. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, page 31–40, New York, NY, USA, 2010. Association for Computing Machinery.

[10] Sumit Bhatia, Purusharth Dwivedi, and Avneet Kaur. That's interesting, tell me more! finding descriptive support passages for knowledge graph relationships. In *The Semantic Web. International Semantic Web Conference*, Lecture Notes in Computer Science, pages 250–267. Springer, Cham, 2018.

[11] Roi Blanco, Harry Halpin, Daniel M Herzig, Peter Mika, Jeffrey Pound, Henry S Thompson, and T Tran Duc. Entity search evaluation over structured web data. In *Proceedings of the 1st International Workshop on Entity-oriented Search*, volume 14 of *SIGIR '11*, pages 2181–2187, New York, NY, USA, 2011. Association for Computing Machinery.

[12] Roi Blanco, Harry Halpin, Daniel M. Herzig, Peter Mika, Jeffrey Pound, Henry S. Thompson, and Thanh Tran. Repeatable and reliable semantic search evaluation. *Web Semantics*, 21:14–29, August 2013.

[13] Roi Blanco and Hugo Zaragoza. Finding support sentences for entities. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 339–346, New York, NY, USA, 2010. Association for Computing Machinery.

[14] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.

[15] Marc Bron, Krisztian Balog, and Maarten de Rijke. Example based entity search in the web of data. In *Advances in Information Retrieval, Proceedings of the 35th European Conference on IR Research (ECIR 2013)*, Lecture Notes in Computer Science, pages 392–403, Berlin, Heidelberg, 2013. Springer.

[16] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[17] Shubham Chatterjee and Laura Dietz. Why does this entity matter? support passage retrieval for entity retrieval. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '19, page 221–224, New York, NY, USA, 2019. Association for Computing Machinery.

[18] Shubham Chatterjee and Laura Dietz. Entity retrieval using fine-grained entity aspects. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1662–1666, New York, NY, USA, 2021. Association for Computing Machinery.

[19] Jing Chen, Chenyan Xiong, and Jamie Callan. An empirical study of learning to rank for entity search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 737–740, New York, NY, USA, 2016. Association for Computing Machinery.

[20] Peter Pin-Shan Chen. The entity-relationship model&mdash;toward a unified view of data. *ACM Transaction Database Systems*, 1(1):9–36, March 1976.

[21] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733, 2016.

[22] Marek Ciglan, Kjetil Nørvåg, and Ladislav Hluchý. The semsets model for ad-hoc semantic list search. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 131–140, New York, NY, USA, 2012. Association for Computing Machinery.

[23] W Bruce Croft and David J Harper. Using probabilistic models of document retrieval without relevance information. *Journal of documentation*, 35(4):285–295, 1979.

[24] Hongliang Dai, Donghong Du, Xin Li, and Yangqiu Song. Improving fine-grained entity typing with entity linking. 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)*, pages 6210–6215, Hong Kong, China, November 2019. Association for Computational Linguistics.

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

[26] Jeffrey Dalton, Laura Dietz, and James Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, page 365–374, New York, NY, USA, 2014. Association for Computing Machinery.

[27] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[28] Arjen P De Vries, Anne-Marie Vercoustre, James A Thom, Nick Craswell, and Mounia Lalmas. Overview of the inex 2007 entity ranking track. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 245–251. Springer, 2007.

[29] Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. Why finding entities in wikipedia is difficult, sometimes. *Information Retrieval*, 13(5):534–567, October 2010.

[30] Gianluca Demartini, Tereza Iofciu, and Arjen P. de Vries. Overview of the inex 2009 entity ranking track. In *Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2009)*, Lecture Notes in Computer Science, pages 254–264, Berlin, Heidelberg, 2009. Springer.

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[32] Laura Dietz. Ent rank: Retrieving entities for topical information needs through entity-neighbor-text relations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 215–224, New York, NY, USA, 2019. Association for Computing Machinery.

[33] Laura Dietz and John Foley. Trec car y3: Complex answer retrieval overview. In *Proceedings of Text REtrieval Conference (TREC)*, 2019.

[34] Laura Dietz, Ben Gamari, Jeff Dalton, and Nick Craswell. Trec complex answer retrieval overview. In *Proceedings of Text REtrieval Conference (TREC)*, 2018.

[35] Laura Dietz, Michael Schuhmacher, and Simone Paolo Ponzetto. Queripidia: Query-specific wikipedia construction. *Proceedings of the Automatic Knowledge Base Construction (AKBC) Workshop*, 2014.

[36] Pranay Dugar. Attention — seq2seq models. `https://towardsdatascience.com/day-1-2-attention-seq2seq-models-65df3f49e263`, June 2019.

[37] Jesse Dunietz and Daniel Gillick. A new entity salience task with millions of training examples. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 205–209, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[38] Ofer Egozi, Evgeniy Gabrilovich, and Shaul Markovitch. Concept-based feature generation and selection for information retrieval. In *AAAI*, volume 2, pages 1132–1137, 2008.

[39] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Trans. Inf. Syst.*, 29(2), April 2011.

[40] Faezeh Ensan and Ebrahim Bagheri. Document retrieval model through semantic linking. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, WSDM '17, page 181–190, New York, NY, USA, 2017. Association for Computing Machinery.

[41] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1625–1628, New York, NY, USA, 2010. Association for Computing Machinery.

[42] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1625–1628, New York, NY, USA, 2010. Association for Computing Machinery.

[43] Besnik Fetahu, Katja Markert, and Avishek Anand. Automated news suggestions for populating wikipedia entity pages. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 323–332, New York, NY, USA, 2015. Association for Computing Machinery.

[44] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, page 363–370, USA, 2005. Association for Computational Linguistics.

[45] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, pages 1301–1306, 2006.

[46] Evgeniy Gabrilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34:443–498, 2009.

[47] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[48] Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. Modeling interestingness with deep neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2–13, Doha, Qatar, October 2014. Association for Computational Linguistics.

[49] Darío Garigliotti and Krisztian Balog. On type-aware entity retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '17, page 27–34, New York, NY, USA, 2017. Association for Computing Machinery.

[50] Emma J Gerritse, Faegheh Hasibi, and Arjen P de Vries. Graph-embedding empowered entity retrieval. In *Advances in Information Retrieval, Proceedings of the 42nd European Conference on Information Retrieval (ECIR 2020)*, Lecture Notes in Computer Science, pages 97–110, Cham, 2020. Springer.

[51] Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820, 2014.

[52] David Graus, Manos Tsagkias, Wouter Weerkamp, Edgar Meij, and Maarten de Rijke. Dynamic collective entity representations for entity ranking. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, page 595–604, New York, NY, USA, 2016. Association for Computing Machinery.

[53] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 267–274, New York, NY, USA, 2009. Association for Computing Machinery.

[54] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, page 209–218, New York, NY, USA, 2016. Association for Computing Machinery.

[55] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1265–1268, New York, NY, USA, 2017. Association for Computing Machinery.

[56] Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. WikiAsp: A Dataset for Multi-domain Aspect-based Summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225, 03 2021.

[57] Rani Horev. Bert explained: State of the art language model for nlp. *Towards Data Science*, 2018.

[58] Lifu Huang, Jonathan May, Xiaoman Pan, and Heng Ji. Building a fine-grained entity typing system overnight for a new X (X = language, domain, genre). *CoRR*, abs/1603.03112, 2016.

[59] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China, July 2015. Association for Computational Linguistics.

[60] K Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information processing & management*, 36(6):809–840, 2000.

[61] Amina Kadry and Laura Dietz. Open relation extraction for support passage retrieval: Merit and open issues. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1149–1152, New York, NY, USA, 2017. Association for Computing Machinery.

[62] Rianne Kaptein and Jaap Kamps. Exploiting the category structure of wikipedia for entity ranking. *Artificial Intelligence*, 194:111–129, January 2013.

[63] Rianne Kaptein, Pavel Serdyukov, Arjen De Vries, and Jaap Kamps. Entity ranking using wikipedia as a pivot. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 69–78, New York, NY, USA, 2010. Association for Computing Machinery.

[64] Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`, May 2015.

[65] Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4999–5007, 2017.

[66] Samia Khalid. Bert explained: A complete guide with theory and tutorial. *Medium*, 2019.

[67] Jinyoung Kim, Xiaobing Xue, and W. Bruce Croft. A probabilistic retrieval model for semistructured data. In *Advances in Information Retrieval, Proceedings of the 31st European Conference on Information Retrieval (ECIR 2009)*, Lecture Notes in Computer Science, pages 228–239, Berlin, Heidelberg, 2009. Springer.

[68] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[69] Oren Kurland. The cluster hypothesis in information retrieval. In *Advances in Information Retrieval, Proceedings of the 36th European Conference on IR Research (ECIR 2014)*, Lecture Notes in Computer Science, pages 823–826. Springer, 2014.

[70] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. *SIGIR Forum*, 51(2):260–267, August 2001.

[71] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: BERT and beyond. *CoRR*, abs/2010.06467, 2020.

[72] Thomas Lin, Patrick Pantel, Michael Gamon, Anitha Kannan, and Ariel Fuxman. Active objects: Actions for entity-centric search. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 589–598, New York, NY, USA, 2012. Association for Computing Machinery.

[73] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015.

[74] Xiao Ling and Daniel Weld. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, volume 26, pages 94–100, September 2012.

[75] Xitong Liu, Fei Chen, Hui Fang, and Min Wang. Exploiting entity relationship for query expansion in enterprise search. *Information Retrieval*, 17(3):265–294, 2014.

[76] Xitong Liu and Hui Fang. Latent entity space: A novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.

[77] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2395–2405, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[78] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[79] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[80] Jarana Manotumruksa, Jeff Dalton, Edgar Meij, and Emine Yilmaz. Crossbert: A triplet neural architecture for ranking entity properties. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 2049–2052, New York, NY, USA, 2020. Association for Computing Machinery.

[81] Edgar "Meij, Marc Bron, Laura Hollink, Bouke Huurnink, and Maarten" de Rijke. Mapping queries to the linking open data cloud: A case study using dbpedia. *Journal of Web Semantics*, 9(4):418 – 433, 2011. JWS special issue on Semantic Search.

[82] Edgar Meij, Dolf Trieschnigg, Maarten de Rijke, and Wessel Kraaij. Conceptual language models for domain-specific retrieval. *Information Processing and Management*, 46(4):448 – 469, 2010. Semantic Annotations in Information Retrieval.

[83] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, page 1–8, New York, NY, USA, 2011. Association for Computing Machinery.

[84] Donald Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Inf. Retr.*, 10(3):257–274, June 2007.

[85] Donald Metzler and W. Bruce Croft. 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*, SIGIR '05, page 472–479, New York, NY, USA, 2005. Association for Computing Machinery.

[86] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the 2013 International Conference on Learning Representations*, 2013.

[87] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[88] Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, 13(1):1–126, 2018.

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

[90] Kriz Moses. Encoder-decoder seq2seq models, clearly explained!! `https://medium.com/analytics-vidhya/encoder-decoder-seq2seq-models-clearly-explained-c34186fbf49b`, March 2021.

[91] Federico Nanni, Simone Paolo Ponzetto, and Laura Dietz. Entity-aspect linking: Providing fine-grained semantics of entities in context. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, JCDL '18, page 49–58, New York, NY, USA, 2018. Association for Computing Machinery.

[92] Fedor Nikolaev, Alexander Kotov, and Nikita Zhiltsov. Parameterized fielded term dependence models for ad-hoc entity retrieval from knowledge graph. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 435–444, New York, NY, USA, 2016. Association for Computing Machinery.

[93] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019.

[94] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424, 2019.

[95] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424, 2019.

[96] Paul Ogilvie and Jamie Callan. Combining document representations for known-item search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, page 143–150, New York, NY, USA, 2003. Association for Computing Machinery.

[97] Christopher Olah. Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`, August 2015.

[98] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(4):694–707, apr 2016.

[99] Kishore Papineni. Why inverse document frequency? In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[100] Jovan Pehcevski, James A. Thom, Anne-Marie Vercoustre, and Vladimir Naumovski. Entity ranking in wikipedia: Utilising categories, links and topic difficulty prediction. *Inf. Retr.*, 13(5):568–600, October 2010.

[101] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[102] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[103] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. 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)*, pages 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics.

[104] Francesco Piccinno and Paolo Ferragina. From tagme to wat: A new entity annotator. In *Proceedings of the First International Workshop on Entity Recognition and Disambiguation*, ERD '14, page 55–62, New York, NY, USA, 2014. Association for Computing Machinery.

[105] Giuseppe Pirrò. Explaining and suggesting relatedness in knowledge graphs. In *The Semantic Web. International Semantic Web Conference*, Lecture Notes in Computer Science, pages 622–639. Springer, Cham, 2015.

[106] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online, November 2020. Association for Computational Linguistics.

[107] Jay Michael Ponte and W Bruce Croft. *A language modeling approach to information retrieval*. PhD thesis, University of Massachusetts at Amherst, 1998.

[108] Marco Ponza, Paolo Ferragina, and Soumen Chakrabarti. A two-stage framework for computing entity relatedness in wikipedia. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1867–1876, New York, NY, USA, 2017. Association for Computing Machinery.

[109] Marco Ponza, Paolo Ferragina, and Francesco Piccinno. Swat: A system for detecting salient wikipedia entities in texts. *Computational Intelligence*, 04 2018.

[110] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 771–780, New York, NY, USA, 2010. Association for Computing Machinery.

[111] Jordan Ramsdell and Laura Dietz. A large test collection for entity aspect linking. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, page 3109–3116, New York, NY, USA, 2020. Association for Computing Machinery.

[112] Hadas Raviv, David Carmel, and Oren Kurland. A ranking framework for entity oriented search using markov random fields. In *Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search*, JIWES '12, New York, NY, USA, 2012. Association for Computing Machinery.

[113] Hadas Raviv, Oren Kurland, and David Carmel. Document retrieval using entity-based language models. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 65–74, New York, NY, USA, 2016. Association for Computing Machinery.

[114] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. Document filtering for long-tail entities. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 771–780, New York, NY, USA, 2016. Association for Computing Machinery.

[115] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.

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

[117] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, page 42–49, New York, NY, USA, 2004. Association for Computing Machinery.

[118] Stephen E Robertson and K Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

[119] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[120] Livia Ruback, Claudio Lucchese, Alexander Arturo Mera Caraballo, Grettel Monteagudo García, Marco Antonio Casanova, and Chiara Renso. Computing entity semantic similarity by features ranking. *arXiv preprint arXiv:1811.02516*, 2018.

[121] Andrew Runge and Eduard Hovy. Exploring neural entity representations for semantic information. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 204–216, Online, November 2020. Association for Computational Linguistics.

[122] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[123] Michael Schuhmacher, Laura Dietz, and Simone Paolo Ponzetto. Ranking entities for web queries through text and knowledge. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 1461–1470, New York, NY, USA, 2015. Association for Computing Machinery.

[124] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, page 101–110, New York, NY, USA, 2014. Association for Computing Machinery.

[125] Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain, April 2017. Association for Computational Linguistics.

[126] Ian Soboroff, Arjen P de Vries, and Nick Craswell. Overview of the trec 2006 enterprise track. In *Trec*, volume 6, pages 1–20, 2006.

[127] Ian Soboroff, Shudong Huang, and Donna Harman. Trec 2018 news track overview. In *TREC*, 2018.

[128] Karen Sparck Jones. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR, 1988.

[129] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[130] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 2–10, New York, NY, USA, 1998. Association for Computing Machinery.

[131] Alberto Tonon, Gianluca Demartini, and Philippe Cudré-Mauroux. Combining inverted indices and structured search for ad-hoc object retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 125–134, New York, NY, USA, 2012. Association for Computing Machinery.

[132] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[133] Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. *REL: An Entity Linker Standing on the Shoulders of Giants*, page 2197–2200. Association for Computing Machinery, New York, NY, USA, 2020.

[134] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[135] Nikos Voskarides, Edgar Meij, and Maarten de Rijke. Generating descriptions of entity relationships. In *Advances in Information Retrieval. European Conference in Information Retrieval*, Lecture Notes in Computer Science, pages 317–330. Springer, Cham, 2017.

[136] Nikos Voskarides, Edgar Meij, Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. Learning to explain entity relationships in knowledge graphs. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 564–574, Beijing, China, July 2015. Association for Computational Linguistics.

[137] Qiuyue Wang, Jaap Kamps, Georgina Ramirez Camps, Maarten Marx, Anne Schuth, Martin Theobald, Sairam Gurajada, and Arunav Mishra. Overview of the inex 2012 linked data track. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[138] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A unified model for knowledge embedding and pretrained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

[139] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press, 2014.

[140] Lilian Weng. Attention? attention! *lilianweng.github.io*, 2018.

[141] Ian H Witten and David N Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. 2008.

[142] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2659–2665. AAAI Press, 2016.

[143] Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Improving neural fine-grained entity typing with knowledge attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, April 2018.

[144] Chenyan Xiong and Jamie Callan. Esdrank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, page 951–960, New York, NY, USA, 2015. Association for Computing Machinery.

[145] Chenyan Xiong and Jamie Callan. Query expansion with freebase. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, page 111–120, New York, NY, USA, 2015. Association for Computing Machinery.

[146] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Bag-of-entities representation for ranking. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, page 181–184, New York, NY, USA, 2016. Association for Computing Machinery.

[147] Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 763–772, New York, NY, USA, 2017. Association for Computing Machinery.

[148] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 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*, SIGIR '17, page 55–64, New York, NY, USA, 2017. Association for Computing Machinery.

[149] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Eduard Hovy. Jointsem: Combining query entity linking and entity based document ranking. In *Proceedings of the 2017 ACM SIGIR Conference on Information and Knowledge Management*, CIKM '17, page 2391–2394, New York, NY, USA, 2017. Association for Computing Machinery.

[150] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. Towards better text understanding and retrieval through kernel entity salience modeling. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, page 575–584, New York, NY, USA, 2018. Association for Computing Machinery.

[151] Chenyan Xiong, Russell Power, and Jamie Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1271–1279, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[152] Chenyan Xiong, Russell Power, and Jamie Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1271–1279, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee.

[153] Peng Xu and Denilson Barbosa. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 16–25, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[154] Yang Xu, Gareth J.F. Jones, and Bin Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 59–66, New York, NY, USA, 2009. Association for Computing Machinery.

[155] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30. Association for Computational Linguistics, 2020.

[156] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259, Berlin, Germany, August 2016. Association for Computational Linguistics.

[157] Ikuya Yamada, Hiroyuki Shindo, and Yoshiyasu Takefuji. Representation learning of entities and documents from knowledge base descriptions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 190–201, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[158] Weixin Zeng, Jiuyang Tang, and Xiang Zhao. Measuring entity relatedness via entity and text joint embedding. *Neural Processing Letters*, 50(2):1861–1875, 2019.

[159] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics.

[160] Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. 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*, SIGIR '15, page 253–262, New York, NY, USA, 2015. Association for Computing Machinery.