Winter 2020

# Why Does This Entity Matter? Finding Support Passages for Entities in Search

Shubham Chatterjee
*University of New Hampshire, Durham*

# WHY DOES THIS ENTITY MATTER? FINDING SUPPORT PASSAGES FOR ENTITIES IN SEARCH

BY

SHUBHAM CHATTERJEE

MSc Computer Science, University of Calcutta, India, 2017

THESIS

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science

in

Computer Science

December, 2020

This thesis has been examined and approved in partial fulfillment of the requirements for the degree of Master in Science in Computer Science by:

Thesis Director, Laura Dietz, Assistant Professor
Department of Computer Science

Marek Petrik, Assistant Professor
Department of Computer Science

Elizabeth Varki, Associate Professor and Graduate Program Coordinator
Department of Computer Science

On April 24, 2020

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

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## ABSTRACT

WHY DOES THIS ENTITY MATTER? FINDING SUPPORT PASSAGES FOR ENTITIES IN SEARCH

by

Shubham Chatterjee

University of New Hampshire, December, 2020

In this work, we propose a method to retrieve a human-readable explanation of how a retrieved entity is connected to the information need, analogous to search snippets for document retrieval. Such an explanation is called a support passage.

Our approach is based on the idea: a good support passage contains many entities relevantly related to the target entity (the entity for which a support passage is needed). We define a relevantly related entity as one which (1) occurs frequently in the vicinity of the target entity, and (2) is relevant to the query. We use the relevance of a passage (induced by the relevantly related entities) to find a good support passage for the target entity. Moreover, we want the target entity to be central to the discussion in the support passage. Hence, we explore the utility of entity salience for support passage retrieval and study the conditions under which it can help. We show that our proposed method can improve performance as compared to the current state-of-the-art for support passage retrieval on two datasets from TREC Complex Answer Retrieval.

# CHAPTER 1

# PRELIMINARIES

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. With the smartphones becoming ubiquitous, we have increasingly come to depend on search functionality to find contacts, email, notes, calendar entries, apps, etc. The field of Information Retrieval (IR) is concerned with developing technology for matching *information needs* with *information objects*. According to Manning et al. [1],

> ***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 query, i.e., the information need, 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. Traditional document retrieval models such as Term Frequency Inverse Document Frequency(TF-IDF) [2–6], BM25 [7] and Language Models [8] are term based models and do not have any notion of semantics in them. For example, TF-IDF is a statistical measure used to evaluate how important a word is to a document in a collection or 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. Similary, BM25 is a *bag-of-words*(text represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity) retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document whereas Language Models are probability distributions over sequences of words where a separate language model is associated with each document in a collection and documents are ranked based on the probability of the query $Q$ in the document's language model $M_d$ ($P(Q|M_d)$. None of these models consider the semantic relationship between various places, events, organizations, etc. in the query or the document.

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*. Analysis of web search query logs has shown that a large portion of the queries now contain some entity, 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. 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. In this report, we provide a brief overview of the existing methods in literature for leveraging entities for passage retrieval. We then describe our current work in progress on explaining query-entity relationships and then using these explanations to derive a better passage ranking.

### 1.1 What is an Entity?

Informally, we call an entity as a "thing" or "object" that one can refer to such as people, locations, products, organizations, and events. However, consider the entity *Apple*. Does this refer to the fruit or the company? 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 [9] defines an entity as follows, taking inspiration from the Entity-

Relationship (ER) Model proposed by Chen [10] 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 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*.

### 1.3 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 [9]:

> ***Definition 4:*** A *Knowledge Repository* (KR) 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). 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 [9]:

***Definition 5:*** A *Knowledge Base* (KB) 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).

# CHAPTER 2
# INTRODUCTION TO SUPPORT PASSAGE RETRIEVAL

Search engines have become ubiquitous in the present world, and search engines which rank entities are integrated into large-scale commercial services such as Facebook (which allows us to search for people), Amazon (which allows us to search for products), etc. 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. However, the entity ranking systems lack the "snippet retrieval" feature which is ubiquitous in document ranking systems. Search snippets play an important role in guiding users to the right documents [11]. Large-scale knowledge bases (such as Freebase and DBpedia) contain facts about entities such as their attributes and relations to other entities. While retrieval of entities from knowledge graphs 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 is not a meaningful explanation of relevance [12].

Several studies show that 40-70% of all web searches target entities [9, 13]. The information need may be a factoid question such as Who is the Prime Minister of the UK? [1] which requires the response to consist of only one entity, or a short information need which requires the retrieval of all topically related entities. Such short information needs are best answered by giving the user a ranking of relevant entities. Entity ranking as a task has been extensively studied in the past [14–18] and several applications display a ranking of entities for a

[1] In this paper, we use the Computer Modern font for queries and Latin Modern Sans font for entities.

user information need. For example, TextMed [2] is a search engine which displays a ranking of entities for a medical information need such as Diabetes, whereas the search engine on Amazon [3] displays a ranking of entities for an information need such as Best Cameras for YouTube Videos.

For information needs such as the ones above, entities are retrieved and ranked according to their relevance to the given information need. Many times, the reason for the relevance, that is, the relationship between the information need and the retrieved entity may not be apparent from the ranking. In such cases, it may be more useful to present a short text snippet explaining how the retrieved entity is related to the information need along with the retrieved entity. As an example, consider the medical information need Diabetes mentioned above. Using TextMed displays the ranking of entities as shown in Figure 2.1. However, for a user unfamiliar with the medical domain, it is not clear from this ranking *how* or *why* the entity Hypertension at rank 1 is related to the information need Diabetes. In such a scenario, displaying a short text snippet such as the one shown in Figure 2.1 can help to clarify the relationship between Diabetes and Hypertension. It may also help the user decide if this entity is of interest to them. As another example, in Figure 2.2, the passage explains how the entity Narendra Modi is affecting the pandemic situation in India through his new policy related to the pandemic, and hence provides an explanation of *how* or *why* Narendra Modi is relevant to the information need COVID-19 in India. Without this supporting passage, a user might not understand the relation of the Prime Minister of India to an ongoing pandemic.

Tombros et al. [11] 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 passage to the user which explains why the entity is relevant to the information need.

In this regard, it is important to note that it has been shown by Dietz et al. [12] that in less than 50% cases, the entity description from a knowledge base or the Wikipedia article of the

[2] `http://www.textmed.com/`
[3] `http://www.amazon.com/`

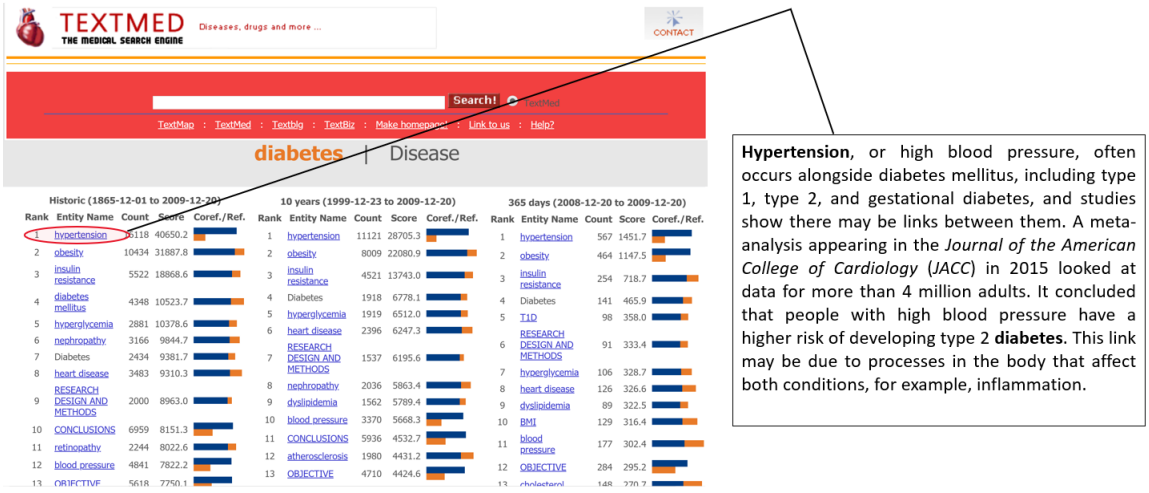



Figure 2.1: An example support passage for the entity Hypertension relevant to the information need Diabetes. This support passage explains how the entity is related to the information need. Without this passage, the entity ranking does not make much sense to a person who does not have knowledge about Hypertension and Diabetes.

entity are useful as explanations of entity relevance to query. Similarly, the organizers of the TREC Complex Answer Retrieval (CAR) [19] track found that the lead paragraph from the Wikipedia article of an entity is not a good explanation for the relevance of the entity, given the query. The participants at the entity retrieval task of TREC Complex Answer Retrieval (CAR) track were asked to submit entity rankings for a query, along with passages from Wikipedia which explain how the entity is related to the query. However, not all participants submitted results with entity explanations. Hence, during assessment, the assessors were provided with the lead paragraph from the Wikipedia article of the corresponding entity. It was found that the lead paragraph from the Wikipedia article of the entity was generally not relevant [19].

**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*.

The importance of this task is also shown by the fact that recently, entity support passage

retrieval has been the subject of various tracks at conferences such as the Text Retrieval Conference (TREC) and Forum for Information Retrieval Evaluation (FIRE). In particular, the entity retrieval task of TREC Complex Answer Retrieval (CAR) track [20] 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 current edition of TREC News [21] in 2020 [4] offers a *Wikification* task where the goal is to link the entities in text to an external resource such as Wikipedia which provides more information on the entity. The Retrieval From Conversational Dialogues (RCD) [5] track at FIRE 2020 provides a passage retrieval task where given an excerpt of a dialogue, the task is to return a ranked list of passages from Wikipedia containing information on the entities in the dialogue.

Such a support passage retrieval system 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 such as Coronavirus Disease 2019 and present them with a Wikipedia-like article on the topic. In fact, given a complex information need such as the one above, the goal of CAR is the construction of an automated information retrieval system to retrieve, cluster, and summarize, to organize relevant information. Topics (such as Coronavirus Disease 2019) would have several facets (like "symptoms", "diagnosis", "prevention", etc.) which would need to be covered. A system which aims to create a Wikipedia-like article about the topic would (1) retrieve relevant entities and passages, (2) cluster along relevant facets, and (3) summarize each cluster with natural language generation. Here, we focus on the first step, where relevant entities and passages are retrieved. In particular, we focus on the passage retrieval step which retrieves passages relevant to the entity in the context of the query. We envision that such support passages would then be clustered and summarized to generate the Wikipedia-like article on the topic.

The current state-of-the-art for entity support passage retrieval [22, 23] uses methods based on entity statistics such as frequency (number of candidate support passages mention-

[4] `http://trec-news.org/`

[5] `https://rcd2020firetask.github.io/RCD2020FIRETASK/`

**Query:** COVID-19 in India
**Target Entity:** Narendra Modi
**Entity Support Passage:**
On 24 March 2020, the Government of India under Prime Minister Narendra Modi ordered a nationwide lockdown for 21 days, limiting movement of the entire 1.3 billion population of India as a preventive measure against the COVID-19 pandemic in India.

Figure 2.2: Example query and entity with support passage.

ing the target entity), the KL-Divergence between the query and collection distributions, relation extraction, etc. However, for support passage retrieval, it is essential to identify the information about the entity which is relevant in the context of the given query. For example, in Figure 2.2, the entity Narendra Modi has been mentioned in the support passage in the context of his role as the Prime Minister of India. However, in some other passage, he may be mentioned in context of his role as the Chief Minister of Gujrat (Gujrat is a state on the west coast of India). The current state-of-the-art for support passage retrieval does not model this. In this work, we identify the query-relevant entity information in query-relevant passages mentioning the target entity $e_i$ to find support passages $s_{ik}$ for the target entity. For this, we use the other entities which frequently co-occur with the target entity. The hypothesis is that a passage containing many entities which frequently co-occur with the target entity and relevant to the query would also mention the target entity and would be good support passage for the entity. We present a novel model called **Entity Prominence** which uses the other entities which frequently co-occur with the target entity and show that it achieves new state-of-the-art results on the task.

Several entity salience detection methods have been developed in recent years [24–26]. In addition to being relevant for the query $Q$, each support passage $s_{ik}$ should mention the target entity $e_i$ 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, given the query and entity in Figure 2.2, consider the two passages in Figure 2.3. In this figure, Passage 1 discusses how

**Passage 1.** Indian Prime Minister Narendra Modi has extended the country's nationwide lockdown until May 3 in a bid to contain the continued spread of the coronavirus, but said that some states which have avoided outbreaks may be allowed to resume "important activities."
**Passage 2.** Home minister Amit Shah said on Sunday that India, despite being densely populated, had coped well with the Covid-19 crisis under Prime Minister Narendra Modi while the health services of most developed nations collapsed because of the pandemic. He added that there was no sense of panic in India over the outbreak.

Figure 2.3: Two example passages mentioning the entity *Narendra Modi* in context of the query *COVID-19 in India*. Passage 1 is salient whereas Passage 2 is not.

Narendra Modi is affecting the pandemic situation in India whereas Passage 2 just mentions the entity on the side. We say that the entity Narendra Modi is salient in Passage 1 but not in Passage 2. The current state-of-the-art for support passage retrieval does not consider the salience of the target entity in the support passage. Hence, such methods might retrieve Passage 2 in Figure 2.3 as a support passage for the entity Narendra Modi in Figure 2.2 although the entity is not central to the discussion in the passage and does not clarify the relation between the query and the entity. Ideally, Passage 1 would be retrieved as the support passage. We incorporate the salience of the target entity in a candidate support passage. We show that these methods can achieve new state-of-the-art results on the task [6]. We explore the extent to which salience detection can help our task.

**Contributions** The contributions of this work are as follows:

1. We propose a new model for support passage retrieval called Entity Prominence. We show that our method achieves new state-of-the-art results for support passage retrieval by improving retrieval effectiveness by 80% (in terms of Mean Average Precision) on average, on two publicly available datasets.

2. We show that entity salience is a useful indicator and can improve retrieval effectiveness

[6]In our work, we use the entity salience detection system from Ponza et al. [26] to predict the salience of an entity in a given passage due to its superior performance on several datasets and its ease of use via an API.

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 background information used. Using a background information about the target entity which is not related to the query (such as the Wikipedia article of the target entity) can perform well; however, the performance is inferior to using information from passages which are relevant to the query and also mention the target entity.

**Outline.** The remainder of this thesis is organized as follows. Chapter 3 discusses some related work on the topic. Chapter 4 presents our proposed method in detail. Chapter 5 presents a quantitative evaluation of our work. Finally, we conclude the thesis with Chapter 6.

# CHAPTER 3
# RELATED WORK

## 3.1 Support Passage Retrieval

Blanco et al. [22] 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)*, where ranking is done in two ways: (1) using entity scores, and (2) using sentence scores. The sentence scores come from a retrieval model such as BM25. They use several types of entity scores to rank support sentences, such as: (1) sum of retrieval scores of entities in the sentence, and (2) the distance between the last match of query and entity and the length of the sentence. Since their work is the current state-of-the-art for the task, we include it as a baseline in our work.

Kadry et al. [23] use relation extraction using OpenIE for support sentence retrieval. Their work studies whether relation extraction can help in support passage retrieval, and the limitations of the current relation extraction approaches that need to be overcome. As such, most of their features are relation-extraction and NLP based. These features are then used in a learning-to-rank framework.

Blanco et al. use only retrieval-based and entity-based features whereas Kadry et al. mainly focus on whether using relation extraction can help in support passage retrieval. Both do not consider the role of the contextual entities, that is, entities which co-occur with

another entity. In our work, we consider the role of these contextual entities in finding good support passages by incorporating the relatedness of the co-occurring entities to the target entity. Both works do not consider the salience of the entity while finding support passages, nor do they identify query-relevant aspects of the entity. In this work, we incorporate entity salience, and query-relevant entity aspects to find support passages.

### 3.2 Entity Retrieval

Given a keyword query, and an entity catalog $\mathcal{E}$, the ad-hoc entity retrieval task is to return a ranked list of entities in $\mathcal{E}$, ranked by the relevance of each entity to the query [9]. This relevance is inferred from a collection of unstructured and/or semi-structured data. A common approach to solving this problem is to represent each entity as a fielded document using some entity description, and then utilize the extensive body of work on document retrieval. There are two main groups of entity retrieval models: semi-structured models [27–35] and learning-to-rank approaches [36–38].

**Semi-Structured Models.** These models utilize information from a large-scale knowledge repository such as Wikipedia, which contain web pages dedicated to describing entities, to represent the entity as a fielded document. Each field in the document consists of a specific part from the semi-structured data being used, such as title, introductory text, names, etc. Then, document retrieval methods are used to retrieve these document representations of the entities. For example, Kaptein et al. [31] propose to utilize Wikipedia as a pivot for entity ranking by treating each Wikipedia page as an entity. In this case, the title of the page becomes the name of the entity, and the content of the page becomes the entity description. To rank web entities given a query, they first associate target entity types with the query, then rank the Wikipedia pages according to the similarity with the query and the target entity types, and finally find web entities corresponding to the Wikipedia entities.

Balog et al. [30] utilize category information about an entity obtained from a user in a

probabilistic framework, where the query and entity is represented as a tuple consisting of a term-based model and a category-based model, both of which are represented using probability distributions. The entities are ranked using the similarity of these two distributions. Similarly, Meij et al. [27] retrieve an initial candidate set of entities (they refer to them as concepts) using the entity descriptions. Then, a supervised machine learning algorithm is used to classify each candidate entity as relevant or not for the query. On the other hand, Tonon et al. [29] propose a hybrid search system based on two components: (1) an inverted index supporting full text search and, (2) a structured repository to maintain a graph representation of the data. They utilize the inverted index-based search component to retrieve an initial ranked list of entities, which is further refined using the structured repository by selecting new entities or reinforcing the results obtained through the inverted index. More recently, Garigliotti et al. [32] utilize the entity type information by using a generative probabilistic model to rank entities for a query. The query is considered in both the term space and type space.

There have also been approaches in literature which use the Markov Random Field (MRF) [39] model to represent a joint distribution over the terms from an entity's description, and the information from a semi-structured data about the entity. The MRF was originally proposed to model term dependencies for ad-hoc retrieval tasks. This model represents the joint distribution over a set of random variables using an undirected graph, where the nodes represent the random variables and the edges represent dependence semantics between them. Raviv et al. [33] present an MRF-based model to model the various dependencies between the query and entity. An entity is represented using the entity description, entity type and entity name. Then, each of these entity representations is jointly modelled with the query terms using three directed graphs. The first graph models the joint distribution of the entity document with the query terms, the second graph models the joint distribution of the entity type with the query target type, and the third graph models the joint distribution of the entity name with the query terms. The final retrieval score of the entity is is estimated

using a linear aggregation of the scores from the three graphs. More recently, Nikolaev et al. [34] proposed the Parametrized Fielded Sequential Dependence Model (PFSDM) and the Parametrized Fielded Full Dependence Model (PFFDM) as an extension to the Fielded Sequential Dependence Model (FSDM) [28]. The FSDM is an MRF-based entity retrieval model, which takes into account both the term dependencies and the document structure. PFSDM assigns different weights to matches of different fields, query term types, and bia-grams. Unlike PFSDM which accounts for only sequential dependencies between the query terms, PFFDM accounts for all dependencies between the query terms. Hasibi et al. [35] leverage the entity annotations in the queries for entity retrieval. Their method is based on the MRF model wherein they introduce a new component for matching the linked entities from the query.

**Learning-To-Rank Approaches.** These methods also use the information from a semi-structured data; however, they treat them as features for a learning-to-rank system. For example, Schuhmacher et al. [37] utilize the entity links in query-relevant web documents to build on a document retrieval system and an entity linking tool. An initial candidate set of entities for the query is built from the entity links contained in high-ranked documents for the query. This initial candidate set is then re-ranked using a learning-to-rank method, which uses several features based on the entity mention, the interaction of the query and the entity mention, the interaction of the query with the entity, and the relation between the entities in a knowledge base. Graus et al. [36] use the entity description obtained from various sources to represent the entity as fielded documents, where each field corresponds to content from one description source. This is done to address the vocabulary mismatch problem between the queries and entities. Next, a classification-based entity ranker which uses different features is trained to learn weights for these features and combine the content from each field of the entity. More recently, Dietz [38] proposed ENT Rank, a learning-to-rank model which utilizes the information about text for entity retrieval by defining neighbour

relations between entities using the context of the entity. This results in a hypergraph with the entities as nodes and the context-neighbour relations as edges.

In this work, we do not address the entity retrieval task. Rather, we assume that an entity ranking is available as input to our system and we seek to embellish the entities in this ranking with support passages explaining the relationship of the entity to the query. Since our task is to rank passages according to relevance for both, the query and the entity, we reuse some ideas found in the entity retrieval literature to find the relevance of a passage for the entity. In particular, we treat the Wikipedia pages as entities and use the content from the Wikipedia article to derive distributions over the terms and other entities in the article. However, we also incorporate the salience of the target entity in a candidate support passage, and the relatedness of the target entity to the other entities in its context, as well as on its Wikipedia page. The focus is on using entity information for text retrieval.

### 3.3 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 [9]. 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.

**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 $\theta_q$ by using the set of query entities $E_q$. Meij et al. [40] 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. [41] 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. [42] take a linear combination of term scores across multiple entity fields. Meij et al. [40] suggest to sample the terms from documents mentioning the entity if descriptions are not available in the knowledge repository. Dalton et al. [43] 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 [43, 44].

**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. [45] 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 [45–48], one could use any knowledge repository where there is sufficient coverage of concepts and concepts are associated with textual descriptions. Liu et al. [49] 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. [50] 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.

**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 re-

trieval model. For example, Raviv et al. [51] 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. [52] 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. [35] (for entity retrieval) and Xiong et al. [53] (for document retrieval). A line of work by Xiong et al. [53–55] 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 [53]. 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 [54], and (2) Word-Entity Duet (WED) Model [55]. 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.

Although related to the problem of ad-hoc document retrieval using entities in that we too use entity-centric information for text retrieval, our problem is fundamentally different in that we want to model the relevance of a passage for both, the query and the entity. We use the entity information to model the relevance of the passage for the entity, whereas the work described in this section try to model the relevance of a document for a query, using entities in the query and document.

### 3.4 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. Since our work is about explaining query-entity relationships, this task is related to our task. All methods use the relation between the two entities found in some knowledge graph, to find suitable explanations describing those relations.

Several approaches exist in literature to solve this problem. One approach is to treat the problem from a graph perspective and then apply various graph algorithms to it. For example, Pirro et al. [56] considered the problem of explaining how two entities in a knowledge graph might be related 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. [57] rank all the paths between any two entities in a knowledge graph. This can help in explaining relationships between seemingly unconnected entities.

However, Voskarides et al. [58] model the task as a learning-to-rank problem with a rich set of features which include textual, entity and relationship features. Their follow up

work [59] addresses the problem using a template based approach where they first identify representative sentences describing some of the relationship instances type and then identify textual descriptions of other instances of the same relationship type by selecting a suitable template and filling it with appropriate entities. On the other hand, Bhatia et al. [60] address the problem from a probabilistic perspective. Given a passage $p$ and a relation $R$ between two entities, they model the problem using Bayes' Theorem and try to find $P(p \mid R)$.

In this work, our aim is not to explain relations between two entities in a knowledge graph. Rather, given a query, and an entity relevant to the query which is retrieved by an entity retrieval system, we want to explain why or how the entity is relevant to the query. This is different from the entity relation explanation task in that we are not explicitly given the relation between the query and entity, but the support passage retrieval method must infer the implicit relation between the query and entity which makes the entity relevant to the query, as in the case of example shown in Figure 2.2. For the entity relation explanation task, the algorithm assumes that the relation between the two entities is explicitly provided. For example, for the two entities *Donald Trump* and *USA*, the relation *Is_President_Of* is available from the knowledge graph. However, in our example shown in Figure 2.2, such explicit relation definitions are not available and must be inferred from the text.

# CHAPTER 4
# APPROACH

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 (according to an entity ground truth for the query). In this section, we present our proposed methods for entity support passage retrieval in detail. First, we discuss the overarching ideas of our work which underlay all our methods in Section 4.1. Then we discuss our proposed micro-approaches in Sections 4.3 through 4.5. In Chapter 5, we evaluate each micro-approach on its own and in a supervised setting.

## 4.1 Overarching Ideas

Consider again, the query and entity in Figure 2.2. For this query, the entity Narendra Modi is relevant as the Prime Minister of India. However, the entity's role as the Chief Minister of Gujrat is not relevant to the query. Hence, at the heart of our approach is a model which, given a query $Q$ and a target entity $e_T$ relevant to the query, finds a passage $p$ which is relevant to $e_T$ in the context of $Q$. By relevant, we mean that $p$ mentions $e_T$ in a context which is relevant to $Q$. For example, for the example query and entity above, a good support passage would not only mention the entity Narendra Modi but also mention it in the context of its role as the Prime Minister of India.

To find such passages which are relevant to the target entity $e_T$, we need the background information on the target entity. We define this background information as passages which

mention (contain a link to) the target entity. However, as noted above, this background information must be query-specific. We find this query-specific background information on the target entity in two steps: (1) Retrieve passages for $Q$ using a corpus of Wikipedia passages (discussed in Section 5.2.4), and (2) Use the relevant passages (to $Q$) which also mention the target entity $e_T$ as the query-specific background information of $e_T$ (discussed in Section 4.1.1). We refer to this query-specific background information about $e_T$ as the *local context* of $e_T$. Since this local context of $e_T$ contains only passages which are relevant to $Q$ and which mention $e_T$, our assumption is that this local context would also contain a good support passage for $e_T$. We use the terms and entities from the local context of $e_T$ to find good support passages for $e_T$.

We also want $e_T$ to be *salient* to the discussion in the support passage. Hence, we propose some methods which incorporate the salience of $e_T$ in the ranking method to find good support passages.

We note that the background information about the target entity may also be obtained using the Wikipedia article of the target entity. However, this background information would not be query-specific, i.e., passages on the Wikipedia page of the target entity may or may not be relevant to the query. We refer to this query-independent context of the target entity as the *global context*. As a comparison, we include the results from our experiments using this global context and show that this is not enough to achieve good results on the task. This also motivates the use of the local context for the task.

For very popular topics such as COVID-19, support passage 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. Hence, we frame the problem as a ranking problem.

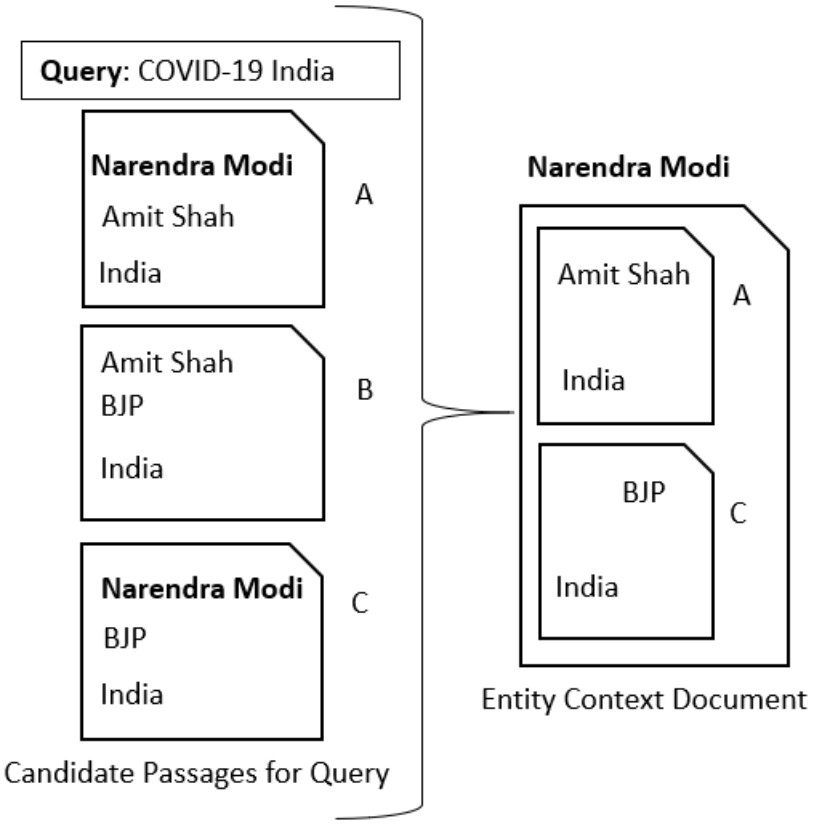



Figure 4.1: Example Entity Context Document (ECD) for query-entity pair shown in Figure 2.2. For the query *COVID-19 in India*, the query-relevant passages mentioning the entity *Narendra Modi* are A and C. These passages are combined together into one composite ECD shown on the right. This ECD contains the candidate support passages (A and C) for the entity *Narendra Modi*, and the entities such as *Amit Shah*, *India*, and *BJP* which co-occur with *Narendra Modi* in it's local context.

### 4.1.1 Constructing the Entity Profile

We construct a representation of the target entity which we refer to as the entity profile [36]. This entity profile serves as the local context of the target entity and provides us with a candidate set of support passages. We use the passages from a candidate passage ranking for the query (Section 5.2.4) which also mention (contain a link to) the target entity to construct this entity profile.

For this, we follow the idea in Dalton et al. [43] and "stitch" all passages which mention the target entity $e_T$ into a composite document $D_{e_T}$. All passages in $D_{e_T}$ mention $e_T$ and are treated as candidate support passages for $e_T$. This is explained in Figure 4.1 for the example

query and entity in Figure 2.2. Our methods for support passage retrieval described below use this entity profile.

### 4.2 Basic Retrieval and Expansion Models

Query expansion [61] is a common technique in information retrieval. It supplements keyword queries with additional terms to get a better sense of the underlying information need. This is often used in conjunction with Pseudo-Relevance Feedback (PRF). In PRF, a set of documents are initially retrieved using the original query. The top-ranked documents from this set are assumed as relevant and expansion terms are derived from these documents. In this work too, we apply query expansion using PRF to expand our queries and retrieve support passages. We use two variations of the Relevance Model (RM) [62] to derive expansion terms for the query:

1. Relevance Model 1 (RM1). In RM1, the probability of a word $w$ given a query $Q$, that is, $P(w \mid Q)$, is estimated by using the query likelihood $P(Q \mid D)$ as the weight for document $D$, and taking an average of the probability of word $w$ given by each document language model $\theta_D$.

2. Relevance Model 3 (RM3). This is similar to RM1 in the estimation of $P(w \mid Q)$. After estimating $P(w \mid Q)$, the relevance model $P(w \mid Q)$ is interpolated with the original query model $\theta_Q$, that is, interpolated with $P(w \mid \theta_Q)$

We expand the queries using both, words and entities. We use the local context of the target entity to derive expansion terms. However, we also include results from replacing the local context with the global context (i.e., Wikipedia article) as previous work has demonstrated the benefits of using external collections for query expansion [63–65].

We experiment with the following retrieval models: BM25, Language Models with Dirichlet Smoothing (LMDS), and Language Models with Jelinek-Mercer Smoothing (LMJM). We

use the Lucene [1] implementation of these retrieval models.

### 4.3 Main Approach: Weighted Entity Prominence (Weighted EPROM)

As discussed in Section 4.1, we use the query-relevant passages which also mention the target entity $e_T$ as the local context of $e_T$ to find the query-relevant background information for $e_T$. To find good explanations of query-entity relations, it is important to model the relevance of the passage for the target entity in the context of the query. For this purpose, we use the entities from the local context of $e_T$.

Our hypothesis is: *A passage containing many entities which are relevant to the query and frequently co-occurring with the target entity mentions the target entity in a query-relevant aspect.* To illustrate this with an example, for the query COVID-19 in India and target entity Narendra Modi, other entities relevant to the query and frequently co-occurring with the target entity might be the entities Amit Shah and India. Our intuition is that a passage which mentions these entities several times would also likely mention Narendra Modi in the correct query-relevant context.

Specifically, we find the frequency with which other query-relevant entities occur in the local context of the target entity. These frequently co-occurring entities may be considered as a measure of the importance of a candidate support passage, with passages which mention many frequently co-occurring entities being more important than others. We refer to this importance measure induced by the co-occurring entities as *Entity Prominence* (EPROM) of a passage, given the query and target entity.

To find most frequently occurring entities in the local context the target entity $e_T$, we use the entity profile $D_{e_T}$ of $e_T$. All entities in the entity profile co-occur with $e_T$. We use the entity profile to derive a distribution over entities $e_d$ which are relevant to $Q$ (according to an entity ground truth) and which co-occur frequently with $e_T$ . We derive this distribution

[1] `https://lucene.apache.org/`

by finding the number of times an entity $e_d$ occurs in the entity profile. More formally,

$$P(e_d \mid e_T, Q) \propto \sum_{p \in D_{e_T}} count(e_d \in p) \tag{4.1}$$

where $p$ is a query-relevant passage mentioning $e_T$, $D_{e_T}$ is the entity profile of the target entity $e_T$, $e_d (\neq e_T)$ is an entity co-occurring with $e_T$ and relevant to $Q$, and $count(e_d)$ is the number of entity links to $e_d$ in $p$.

We define the *Entity Prominence* (EPROM) of a passage in an entity profile $D_{e_T}$ as:

$$\text{EPROM}(p \mid Q, e_T) = \sum_{e_d \in p} P(e_d \mid e_T, Q) \tag{4.2}$$

**Weighted EPROM** We score candidate support passage $p \in D_{e_T}$ by interpolating the entity prominence score from Equation 4.2 with the score of the passage for the query:

$$\text{Score}(p \mid Q, e_T) = \lambda \cdot \text{EPROM}(p \mid Q, e_T) + (1 - \lambda) \cdot \text{Score}_Q(p) \quad \lambda \in [0, 1] \tag{4.3}$$

where $\lambda$ is learnt using a machine learning method. We set $\text{Score}_Q(p)$ equal to the original retrieval score of the passage for the query, obtained from the candidate passage ranking (described in Section 5.2.4).

## 4.4 Other Approaches

**Entity Profile Terms (ProfileTerms)** Analogous to using the frequently co-occurring entities with the target entity in Section 4.3, term statistics may also be used to find passages relevant to the target entity in the context of the query.

For this, we first obtain distribution over the terms $t$ in the entity profile $D_{e_T}$ of the

target entity $e_T$, weighted by the retrieval score of the passage in which it occurs. Formally,

$$P(t \mid e_T, Q) \propto \sum_{p \in D_e} \text{Score}_Q(p) \cdot \text{tf}_p(t) \tag{4.4}$$

where $\text{tf}_p(t)$ is the number of times term $t$ occurs in passage $p$, and $\text{Score}_Q(p)$ is the original retrieval score of the passage $p$ for the query $Q$, obtained from the candidate passage ranking (described in Section 5.2.4).

We then score a candidate support passage by accumulating the term scores of each term in the passage. Formally,

$$\text{Score}(p \mid Q, e_T) = \sum_{t \in p} P(t \mid e_T, Q) \tag{4.5}$$

**Query Expansion Based Methods.** As discussed in Section 3.3, query expansion with pseudo-relevance feedback (PRF) using terms and entities has been successfully applied previously in document retrieval systems. In this work too, we use query expansion with PRF for support passage retrieval. We use two types of expansion units: entities and terms (i.e., words). For every query, we expand the query using terms or entities from the local context of the target entity $e_T$, i.e., the entity profile $D_{e_T}$ of $e_T$. We retrieve support passages using this expanded query from an index consisting of passages from $D_{e_T}$.

1. **Query Expansion using Entities from Entity Profile (QE-Profile-Entities).** We expand the original query using the top 20 co-occurring entities $e_d \in D_{e_T}$ obtained using Equation 4.1.

2. **Query Expansion using Terms from Entity Profile (QE-Profile-Terms).** We expand the original query using the top 50 terms $t \in D_{e_T}$ obtained using Equation 4.4.

For each method above, we experiment with the same variations of retrieval models and expansion models as given in Section 4.2.

### 4.5 Replacing the local context with the global context of the target entity

As noted in Section 4.1, it is also possible to obtain the background information on the target entity using the Wikipedia article of the target entity. However, this background information would not be query-specific. We refer to the Wikipedia article of the target entity as global context of the target entity. To motivate the use of a query-specific local context and the use of our more complicated entity profile (Section 4.1.1) for support passage retrieval, we include results from replacing the local context (entity profile) with the global context (Wikipedia article) in our proposed methods. We study the contribution of the local versus the global context.

1. **WikiTerms.** Similar to our method ProfileTerms described in Section 4.4. Here, we use the Wikipedia article of the target entity to find a distribution over the terms in the Wikipedia article. We do not use the $\text{Score}_Q(p)$ component in Equation 4.5 since the passages in Wikipedia do not have a retrieval score under the query.

2. **WikiEntities.** Similar to EPROM in Section 4.3 described above. However, here the entities $e_w$ come from the Wikipedia article (global context) and not the Entity Profile (local context) $D_{e_T}$ of the target entity $e_T$. We score a candidate support passage $p \in D_{e_T}$ by accumulating (with a sum), the frequency of all entities $e_w \in p$ which are also on the Wikipedia page of $e_T$.

### 4.6 Entity Salience for Support Passage Retrieval

A good support passage must not only mention the target entity but also be about the entity, and must clearly capture how the entity is related to the query. It must be able to answer the question: *What is it about the entity that makes it relevant to the query?* That is, the entity must be central to the discussion in the passage and not just be mentioned in passing. We call an entity as **salient** in some text, if the entity is **central** to the discussion in the

text. For example, the entity *Narendra Modi* from Figure 2.2 is salient in Passage 1 from Figure 2.3 but not in Passage 2.

Although entity salience detection in text has received attention as a stand-alone problem, it is not clear how (or if) entity salience can help in passage retrieval. In this work, we try to bridge this gap by proposing some support passage retrieval methods which take the salience of the entity in the text into consideration and studying if salience is useful for support passage retrieval.

Since the purpose of this work is not to propose a new entity salience detection method but to study if, and how we can use salience for support passage retrieval, we leverage existing work on salience detection. In particular, we use the salience detection system from Ponza et al. [26] as it has been showed to outperform existing state-of-the-art in the field of entity salience detection, and also due to its ease-of-use through as an API.

### 4.6.1 Methods Based on Entity Salience

We denote by Salience($e_T \mid p$), the salience score of the target entity $e_T$ for a support passage $p$. We then score $p$ as follows:

$$\text{Score}(p \mid e_T, Q) = \mu \cdot \text{Salience}(e_T \mid p) \tag{4.6}$$

where $\mu$ is a weight which factors in the relevance of passage and entity respectively, given the query. Hence, we set $\mu$ in two ways:

1. $\mu = \text{Score}(e_T \mid q)$ where $\text{Score}(e_T \mid q)$ is the retrieval score of the entity $e_T$ for the query $Q$ obtained from the input entity ranking (The input entity ranking is described in Section 5.2.5).

2. $\mu = \text{Score}(p \mid q)$ where $\text{Score}(p \mid q)$ is the retrieval score of the passage for the query obtained from the candidate passage ranking. (The candidate passage retrieval for query is described in Section 5.2.4).

We use two sources of candidate support passages $p$.

1. *Candidate passages for the query which also mention the target entity.* To investigate whether entity salience can help us find good support passages, we rank passages from the candidate passage ranking for the query (obtained in Section 5.2.4) which also mention the target entity. To this end, we use the entity profile(Section 4.1.1) for the target entity as a source of support passages for the entity.

2. *Support passages already obtained from a support passage ranking method.* To investigate the effect of entity salience on a support passage ranking, we re-rank support passages obtained using any of our support passage ranking methods. In our experiments, we use the support passages obtained using method *Weighted EPROM* but any support passage ranking method will suffice.

The two settings for $\mu$ and the two candidate support passage sources give us the four combinations of methods based on entity salience below:

1. **Sal-Profile-Psg-Scores.** Salience of target entity in a passage from the entity profile and using score of the passage for the query. In this method, we rank passages from the entity profile using Equation 4.6 with $\mu = \text{Score}(p \mid q)$.

2. **Sal-Profile-Ent-Scores.** Salience of target entity in a passage from the entity profile and using score of the target entity for the query. In this method, we rank passages from the entity profile using Equation 4.6 with $\mu = \text{Score}(e_T \mid q)$.

3. **Sal-SP-Psg-Scores.** Salience of target entity in a passage from a support passage ranking and using score of the passage for the query. In this method, we re-rank passages in a support passage ranking using Equation 4.6 with $\mu = \text{Score}(p \mid q)$.

4. **Sal-SP-Ent-Scores.** Salience of target entity in a passage from a support passage ranking and using score of the target entity for the query. In this method, we re-rank passages in a support passage ranking using Equation 4.6 with $\mu = \text{Score}(e_T \mid q)$.

# CHAPTER 5
# EVALUATION, RESULTS, AND DISCUSSIONS

## 5.1 Research Questions

As discussed in Section 2, it is important to identify the query-relevant information of an entity to find good support passages for the entity. In this work, we rely on the frequently co-occurring entities found in the local context of the target entity, to find query-relevant entity information in a passage. Our hypothesis, as mentioned in Section 4.3, is that a passage containing many entities which frequently co-occur with the target entity, is a good support passage. Hence, the first research question that we aim to answer is:

**RQ1** To what extent are frequently co-occurring entities from the local context of a target entity helpful in support passage retrieval?

The problem of entity salience detection has received attention from the research community; however, it has always been studied separately, and whether or not it can contribute to text retrieval problems has gone answered. In this work, we use the salience of the target entity in the support passage as an indicator of good support passages. With this, the research question we aim to answer is the following:

**RQ2** To what extent is entity salience helpful in support passage retrieval?

As noted in Section 4.1, we may obtain the background information on the target entity in two ways: (1) Query-specific, using query-relevant passages which mention the target entity, and (2) Query-unspecific, using the Wikipedia article of the target entity. We refer

to the former as the local context, and the latter as the global context of the target entity. We study the following research question:

**RQ3** Which background information about the target entity is more useful for support passage retrieval–local or global? Which aspects of that information, terms or entities, contribute more to the overall results?

## 5.2 Evaluation Paradigm

### 5.2.1 Datasets

We use two datasets from the TREC Complex Answer Retrieval (CAR) track [20][1] 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) [66] dataset which consists of questions taken from middle school science curricula. This dataset consists of 27 sections.

### 5.2.2 Input Entity Ranking

Since the input to our methods is an entity ranking, we use an entity ranking obtained using the Wikipedia page titles as queries. We convert TREC CAR title queries into keyword queries by using the page name of the Wikipedia page to construct a boolean query of

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

the terms in the page name. We then retrieve entities from an index containing fielded documents, each representing an entity, using BM25. However, any system could be used to obtain an entity ranking here.

### 5.2.3 Corpus

We use the corpus of passages from TREC CAR. It is an entity linked corpus consisting of paragraphs from the entire English Wikipedia. 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 [67].

### 5.2.4 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.

### 5.2.5 Input Entity Ranking

Since the input to our methods is an entity ranking, we use an entity ranking obtained using the Wikipedia page titles as queries. We convert TREC CAR title queries into keyword queries by using the page name of the Wikipedia page to construct a boolean query of the terms in the page name. We then retrieve entities from an index containing fielded documents, each representing an entity, using BM25. However, any system could be used to obtain an entity ranking here.

### 5.2.6 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. A paragraph is deemed as relevant, if it is contained in the page/section, whereas 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.

### 5.2.7 Knowledge Base

We use Wikipedia as a knowledge base in our work. The TREC CAR dataset consists of a large, unprocessed collection of Wikipedia pages which may be used to derive a knowledge base. It contains all pages except those in the benchmarks. We perceive the knowledge base as text and build a knowledge base index, which associates each entity in the knowledge base with text that includes the Wikipedia article, as well as anchor text, names and type labels.

### 5.2.8 Machine Learning

We apply our methods to produce a support passage ranking for every query-entity pair. We then treat each ranking as a feature and perform 5-fold cross validation with a listwise learning-to-rank (L2R) method (Coordinate Ascent) optimized for Mean Average Precision (MAP). We also use Coordinate Ascent optimized for MAP to set the weights in Equation 4.3.

### 5.2.9 Evaluation Metrics

In this work, we are interested in precision more than recall. This is because although there may be many passages explaining the relationship between a query and an entity, a typical user is interested in one or two of them. Moreover, the user interfaces of entity retrieval systems tend to be very crowded and would typically contain space enough for only one or two such support passages. Hence, we use the following precision-oriented retrieval metrics to evaluate our work: Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision at R (P@R).

### 5.2.10 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 (MAP) of a baseline method, with the 5% most difficult pairs for this method to the left and the 5% easiest ones to the right. Performance scores for the ranking of each query-entity pair are reported as macro-averages, that is, average across all entities first, then average across queries. 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*).

## 5.3 Baselines

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

### 5.3.1 Baselines from Blanco et al.

We re-implement the methods from Blanco et al. [22] and include them as our baselines. Section 3.1 describes their work. Their methods make use of a named entity recognizer to find entities in the candidate support sentences. We use the Stanford Named Entity Recognizer [68] [2] for this purpose. Below, we give a short description of their methods which we include as baselines in this paper.

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

$$\text{Score}_{qe}(p) = \begin{cases} \sum_{e' \in p} E(q, e') & \text{if } e \in p \\ 0 & \text{if } e \notin p \end{cases} \tag{5.1}$$

where $E(q, e')$ is an entity ranking method which scores an entity for the query. Although their formulation uses a summation in the Equation 5.1, in their work, Blanco et al. also

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

experiment with using an average instead of the summation.

Blanco et al. experiment with several entity ranking methods and substitute them for $E(q, e')$ in Equation 5.1. 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,

$$E_{KLD}(q, e) = P(e|\theta_q) \cdot \log \frac{P(e|\theta_q)}{P(e|\theta_C)} \tag{5.2}$$

   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 $e$.

Although their formulation uses a summation in the Equation 5.1, in their work, Blanco et al. also experiment with using an average instead of the summation. In our work too, we include as baselines, the results from using both, an average and a summation in Equation 5.1 with the various entity ranking methods described above. We found that many of these methods have similar performance with no statistical difference and hence we choose to include the ones with combination and KLD as entity ranking methods above. The results on *BenchmarkY1-Train* and *BenchmarkY2-Test* are included in Table 5.1.

### 5.3.2 Other Baselines

In addition to the methods from Blanco et al. we include two additional baselines which use the query and entity, without any other components of our approach. These are:

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 (CompundQuery).** We retrieve passages using a compound query, where the query is a combination of the original query and the target entity.

## 5.4 Results and Discussions

In this section, we discuss each research question presented in Section 5.1.

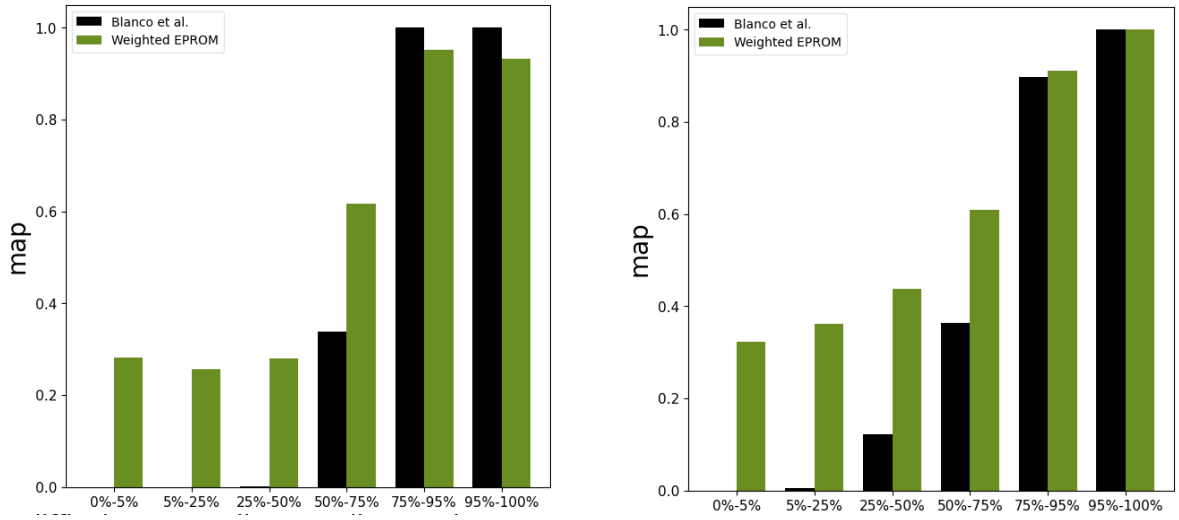
The most interesting results from our support passage retrieval methods on the two datasets are shown in Table 5.1. To study the contribution of the methods in a supervised, learning-to-rank system, we also present results from an ablation study on the two datasets in Table 5.2.

Table 5.1: Performance with standard error of individual support passage ranking methods on BenchmarkY1-Train and BenchmarkY2-Test. The best performing baselines and the best performing methods are in bold.

| | | **BenchmarkY1-Train** | | | **BenchmarkY2-Test** | | |
|---|---|---|---|---|---|---|---|
| **Number** | | **MAP** | **P@R** | **MRR** | **MAP** | **P@R** | **MRR** |
| 1 | FreqOfEntityLinks | 0.16±0.004 | 0.11±0.004 | 0.17±0.004 | 0.19±0.01 | 0.16±0.01 | 0.28±0.01 |
| 2 | Blanco et al. [22] | 0.15±0.004 | 0.13±0.004 | 0.16±0.004 | 0.21±0.01 | 0.20±0.01 | 0.32±0.01 |
| 3 | CompoundQuery | 0.05±0.002 | 0.04±0.002 | 0.05±0.002 | 0.06±0.01 | 0.06±0.01 | 0.12±0.01 |
| 4 | Weighted EPROM | **0.30±0.004** | **0.27±0.004** | **0.33±0.004** | **0.38±0.01** | **0.36±0.01** | **0.50±0.01** |
| 5 | EPROM | 0.28±0.004 | 0.26±0.004 | 0.31±0.004 | 0.32±0.01 | 0.29±0.01 | 0.41±0.01 |
| 6 | ProfileTerms | 0.25±0.004 | 0.21±0.004 | 0.27±0.004 | 0.33±0.01 | 0.30±0.01 | 0.44±0.01 |
| 7 | QE-Profile-Entities (LMJM + RM3) | 0.27±0.004 | 0.24±0.004 | 0.30±0.004 | 0.34±0.01 | 0.31±0.01 | 0.44±0.01 |
| 8 | QE-Profile-Terms (LMJM + RM3) | 0.26±0.004 | 0.22±0.004 | 0.28±0.004 | 0.36±0.01 | 0.33±0.01 | 0.47±0.01 |
| 9 | WikiTerms | 0.24±0.004 | 0.19±0.004 | 0.25±0.004 | 0.31±0.01 | 0.28±0.01 | 0.41±0.01 |
| 10 | WikiEntities | 0.15±0.004 | 0.12±0.004 | 0.17±0.004 | 0.20±0.01 | 0.19±0.01 | 0.31±0.01 |
| 11 | Sal-ECD-Psg-Scores | 0.02±0.003 | 0.02±0.003 | 0.03±0.003 | 0.03±0.01 | 0.04±0.01 | 0.10±0.01 |
| 12 | Sal-ECD-Ent-Scores | 0.02±0.003 | 0.02±0.003 | 0.03±0.003 | 0.03±0.005 | 0.04±0.004 | 0.10±0.005 |
| 13 | Sal-SP-Psg-Scores | 0.02±0.003 | 0.02±0.003 | 0.03±0.003 | 0.02±0.006 | 0.02±0.005 | 0.09±0.006 |
| 14 | Sal-SP-Ent-Scores | 0.02±0.003 | 0.02±0.003 | 0.03±0.003 | 0.02±0.01 | 0.02±0.01 | 0.09±0.01 |

Table 5.2: Learning-To-Rank combination of all features including subsets on BenchmarkY1-Train and BenchmarkY2-Test.

| | | **BenchmarkY1-Train** | | | **BenchmarkY2-Test** | | |
|---|---|---|---|---|---|---|---|
| **Number** | **Subset** | **MAP** | **P@R** | **MRR** | **MAP** | **P@R** | **MRR** |
| 1 | Global Context Features | 0.24±0.004 | 0.19±0.004 | 0.25±0.004 | 0.31±0.01 | 0.27±0.01 | 0.41±0.01 |
| 2 | Local Context Features | 0.30±0.004 | 0.25±0.004 | 0.30±0.004 | 0.35±0.01 | 0.31±0.01 | 0.45±0.01 |
| 3 | Profile Entities | 0.32±0.004 | 0.28±0.004 | 0.34±0.004 | 0.39±0.01 | 0.37±0.01 | 0.51±0.01 |
| 4 | Profile Terms | 0.27±0.004 | 0.23±0.004 | 0.30±0.004 | 0.34±0.01 | 0.30±0.01 | 0.45±0.01 |
| 5 | All Except Profile Entities | 0.30±0.004 | 0.25±0.004 | 0.32±0.004 | 0.38±0.01 | 0.35±0.01 | 0.50±0.01 |
| 7 | **All** | **0.34±0.004** | **0.29±0.004** | **0.36±0.004** | **0.40±0.01** | **0.37±0.01** | **0.52±0.01** |




(a) BenchmarkY1-Train Results.

(b) BenchmarkY2-Test Results.

Figure 5.1: Difficulty-test for MAP, comparing Blanco et al. to our proposed method Weighted EPROM. We observe that for the more difficult query-entity pairs according to the performance of Blanco et al. (0-50% on left), ranking support passages using our method Weighted EPROM can help the task.

### 5.4.1 RQ1: Frequently Co-Occurring Entities

**Difficulty Test.** We observe from Table 5.1 that the state-of-the-art method from Blanco et al. (Row 2) achieves $MAP = 0.15$ on BenchmarkY1-Train and $MAP = 0.21$ on BenchmarkY2-Test. However, our proposed method Weighted EPROM achieves $MAP = 0.30$ on BenchmarkY1-Train, and $MAP = 0.38$ on BenchmarkY2-Test. This gives us an improvement of 100% over Blanco et al. on BenchmarkY1-Train, and 80% on BenchmarkY2-Test, with an average improvement of 90% on the two datasets.

The observations above indicate that frequently co-occurring entities are good indicators of support passages. To investigate the extent to which such co-occurring entities can help the task, we perform the difficulty test explained in Section 5.2.10. We use Blanco et al. [22] as our baseline for comparison, and compare its performance with our method Weighted EPROM. The results on the two datasets are shown in Figure 5.1. We observe that for the more difficult query-entity pairs (according to performance of Blanco et al.), ranking support passages using our proposed method Weighted EPROM helps the task. For example,

we observe in both Figures 5.1a and 5.1b that for the 5% most difficult query-entity pairs (extreme left of the charts), our method Weighted EPROM can find support passages. This supports our hypothesis from Section 4.3 that a support passage mentioning frequently co-occurring entities with the target entity likely also mentions the target entity.

**Helps-Hurts Analysis.** To quantify the discussion above, we also perform the helps-hurts analysis explained in Section 5.2.10 where we analyze how many query-entity pairs were helped by using our method Weighed EPROM as compared to Blanco et al. We found that using Weighted EPROM can help find support passages for 2131 query-entity pairs when compared to Blanco et al. In other words, Blanco et al. cannot find support passages for these query-entity pairs but our proposed method can. This further proves that our initial hypothesis about frequently co-occurring entities is correct.

**Example.** As an example, we show a difficult query-entity pair found during the Helps/Hurts analysis, along with its support passage in Figure 5.2. For this query-entity pair, Blanco et al. was unable to find a support passage; however, our method Weighted EPROM did find one. We found that the top three most frequently co-occurring entities with Genetic Disorder (except itself) are: Gene Therepy, Severe Combined Immunodeficiency and Muscular Dystrophy. We observe in Figure 5.2 that the support passage clarifies that genetic disorders could be treated using genetically modified organisms. Hence, this passage is a good explanation of why the entity Genetic Disorder is relevant to the query Genetically Modified Organism. Our method Weighted EPROM has correctly identified this passage as a support passage since it contains many frequently co-occurring entities with the entity Genetic Disorder and in particular, contains the top three most frequently co-occurring entities. This confirms and supports our hypothesis that a passage mentioning many frequently co-occurring entities with the target entity is a good support passage.

**Ablation Study.** From Table 5.2, we observe that a learning-to-rank system using only methods based on frequently co-occurring entities as features (**Profile Entities**, Subset-

**Query:** Genetically Modified Organism
**Entity:** Genetic Disorder
**Support Passage:**
**Gene therapy**, uses genetically modified viruses to deliver genes that can cure disease in humans. Although gene therapy is still relatively new, it has had some successes. It has been used to treat **genetic disorders** such as **severe combined immunodeficiency**, and Leber's congenital amaurosis. Treatments are also being developed for a range of other currently incurable diseases, such as **cystic fibrosis**, **sickle cell anemia**, **Parkinson's disease**, **cancer**, **diabetes**, **heart disease** and **muscular dystrophy**.

Figure 5.2: Example query and entity with top ranked support passage found by method Weighted EPROM. The frequently co-occurring entities with the entity *Genetic Disorder* found in the passage are in bold.

3) outperforms all baselines in Table 5.1. For example, on BenchmarkY1-Train, **Profile Entities** achieves $MAP = 0.32$ and a learning-to-rank system with all features (*All, Row 7)* achieves $MAP = 0.34$. However, if we remove all the features based on co-occurring entities from our full system, there is a slight drop in performance, from $MAP = 0.34$ on *All* to $MAP = 0.30$ on *Subset-5*. We observe similar results on BenchmarkY2-Test as well.

The observations above further show that frequently co-occurring entities are strong indicators of good support passages. They perform well on their own and outperform the state-of-the-art-baseline for the task. Moreover, they also contribute to the performance of a learning-to-rank-based system which uses other features. This clearly demonstrates the benefits of using frequently co-occurring entities in the support passage retrieval task. We also show the results from a difficulty test, comparing a learning-to-rank system which uses all features except those based on frequently co-occurring entities, to a system which uses all the features, in Figure 5.3. We observe that a system which uses the frequently co-occurring entity-based features can find support passages for even the most difficult 5% of the query-entity pairs. We lose these query-entity pairs when we use a system which does not use the frequently co-occurring entity-based features.

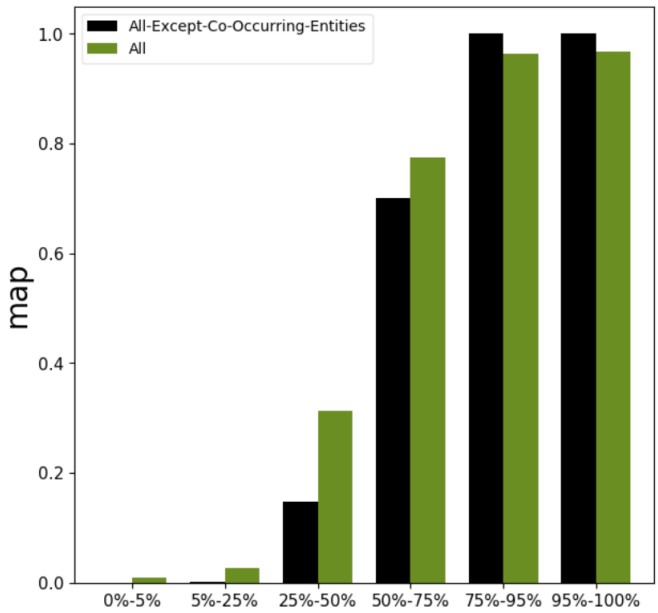



Figure 5.3: Difficulty-test for MAP, comparing a L2R system using all features except those based on co-occurring entities to one which uses all.

**Conclusions.** Regarding **RQ1**, frequently co-occurring entities are beneficial for the support passage retrieval task as good support passages mention many co-occurring entities with the target entity. Using frequently co-occurring entities can help to improve performance over the current state-of-the-art baseline by helping to find support passages for query-entity pairs which are difficult for the baseline. We outperform the current state-of-the-art method for the task using our proposed measure called *entity prominence* (which uses frequently co-occurring entities) on two publicly available benchmarks. Frequently co-occurring entities show their strength by not only performing very well on their own and achieving new state-of-the-art results over several established baselines, but also in a learning-to-rank system which uses several other features.

### 5.4.2 RQ2: Entity Salience

**Observations and Discussions.** From Table 5.1, we observe that retrieving support passages using entity salience performs very poorly. For example, on BenchmarkY1-Train, both methods Sal-Profile-Psg-Scores and Sal-Profile-Ent-Scores achieve a $MAP = 0.02$.

Table 5.3: Results on BenchmarkY1-Train for subset of entities with at least one salient mention.

| | **MAP** | **P@R** | **MRR** |
|---|---|---|---|
| Blanco et al. [22] | **0.14** | **0.12** | **0.19** |
| Weighted EPROM | **0.27** | **0.25** | **0.42** |
| Sal-Profile-Psg-Scores | **0.24** | **0.24** | **0.38** |
| Sal-Profile-Ent-Scores | 0.23 | 0.23 | 0.35 |
| Sal-SP-Psg-Scores | **0.25** | **0.25** | **0.40** |
| Sal-SP-Ent-Scores | 0.22 | 0.22 | 0.35 |

Similarly, on BenchmarkY2-Test, both methods Sal-Profile-Psg-Scores and Sal-Profile-Ent-Scores achieve a $MAP = 0.03$. This is much below the baseline of Blanco et al. which achieves a $MAP = 0.15$.

Moreover, re-ranking support passages using entity salience too performs very poorly. For example, in Table 5.1, on both BenchmarkY1-Train and BenchmarkY2-Test, both methods Sal-SP-Psg-Scores and Sal-SP-Ent-Scores achieve a $MAP = 0.02$. This is much below the method Weighted EPROM [3] which achieves $MAP = 0.30$.

The observations above indicate that entity salience is not helping the support passage retrieval task. A helps/hurts analysis shows that as compared to baseline Blanco et al., Sal-ECD-Psg-Scores helps 237 but hurts 635 query-entity pairs. Similarly, as compared to Weighted EPROM, Sal-SP-Psg-Scores helps 154 but hurts 1118 query-entity pairs.

We manually confirmed that the system SWAT [26] correctly identifies salient and non-salient entities; however, *only few retrieved entities have a passage with a salient mention in the candidate set*. While entities with salient passages are often relevant, a majority (95%) of retrieved entities *do not* have a passage with a salient mention in the candidate pool. Since the salience is only applicable to very few entities, it only has a limited impact on the overall result.

To study whether salience is a useful indicator when it is applicable, we analyze results

[3] We compare Sal-SP-Psg-Scores and Sal-SP-Ent-Scores to Weighted EPROM because our results for both methods are obtained by re-ranking the support passages obtained using Weighted EPROM. However, any support passage retrieval method will suffice.

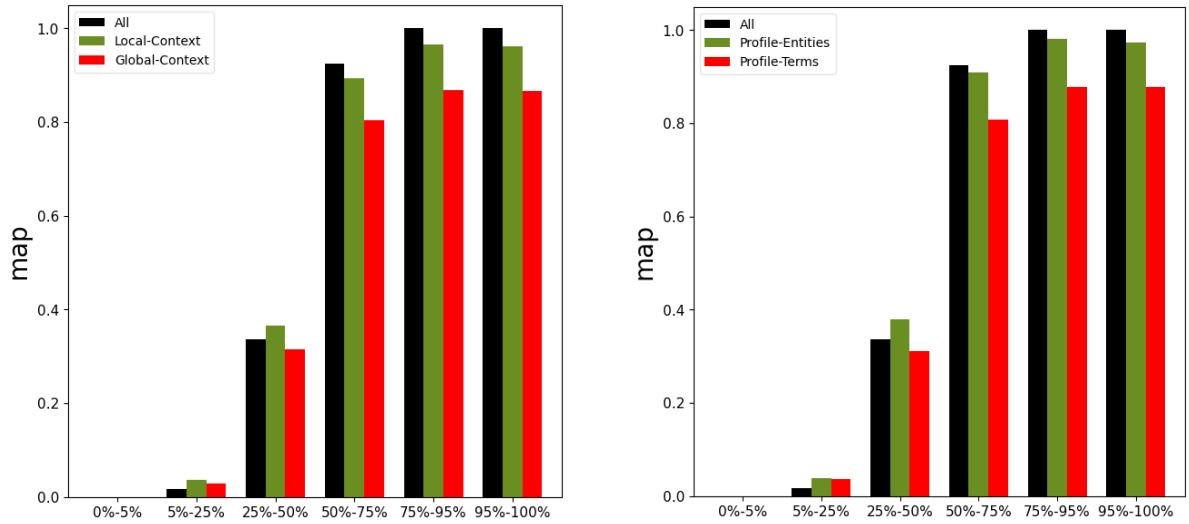
on the subset of rankings for query-entity pairs for which the passage ranking contains at least one passage in which the target entity is salient. The results on BenchmarkY1-Train are shown in Table 5.3. We now observe that Sal-Profile-Psg-Scores with $MAP = 0.24$ outperforms Blanco et al. with $MAP = 0.15$. This is an improvement of 60% over Blanco et al. in terms of Mean Average Precision. Moreover, Sal-Profile-Psg-Scores has performance only slightly worse than that of Weighted EPROM. Hence, salience is a useful indicator; however, it is only applicable for entities which have a salient passage in the candidate pool.

**Conclusions.** With respect to **RQ3**, we conclude that entity salience is a useful indicator of support passages. However, many entities do not have a passage with a salient mention in the candidate set and hence salience is not applicable to these entities. This hurts the performance of a learning-to-rank system using entity salience as a feature. However, whenever applicable, entity salience can help improve performance over the state-of-the-art.

### 5.4.3 RQ3: Local Context Versus Global Context

**Global Versus Local Context.** From Table 5.1, on both datasets, we observe that the performance of method WikiTerms (Row 9) which uses terms from the global context of the target entity, and ProfileTerms (Row 6) which uses the terms from the local context of the target entity are similar. However, EPROM (Row 5) which uses entities from the local context outperforms WikiEntities (Row 10) which uses entities from the global context by a huge margin. To investigate further, we present results from an ablation study in Table 5.2.

From Table 5.2, we observe that a learning-to-rank system using only local context features (Subset-2) outperforms the system using only global context features (Subset-1). For example, on BenchmarkY1-Train, *Subset-2* achieves $MAP = 0.30$ compared to *Subset-1*, which achieves $MAP = 0.24$. Similarly, on BenchmarkY2-Test, *Subset-2* achieves $MAP = 0.35$ compared to *Subset-1*, which achieves $MAP = 0.31$. Both systems (*Subset-1* and *Subset-2*) outperform all baselines in Table 5.1. This shows that although global infor-




(a) Difficulty test to determine the importance of knowledge base information versus contextual information.

(b) Difficulty test to determine the importance of entities from profile of target entity versus terms from profile of target entity.

Figure 5.4: Difficulty test for MAP, comparing different L2R systems. Difficulty percentile is according to performance of **All**.

mation from the Wikipedia article of the target entity is a strong indicator of good support passages, they are less informative than the local information from the ECD of the target entity.

To verify the above, we also perform a difficulty test, comparing the learning-to-rank systems consisting of features based on only global context and only local context of the target entity respectively, to a system consisting of all features. The results for BenchmarkY1-Train are shown in Figure 5.4a. The results for BenchmarkY2-Test are similar and hence omitted for brevity. We observe that whenever the system *All* finds it difficult to predict support passages for some query-entity pairs, the local and global features help to improve the mean statistics. However, the contribution of the local context is always more than that of the global context.

**Terms Versus Entities.** From the discussion above, we may conclude that information from local context is more important and informative than that from global context of the target entity. However, which local contextual information is more important – Terms or

Entities? From Table 5.2, we observe that a learning-to-rank system consisting of only profile entity features outperforms that consisting of only profile terms features. For example, on BenchmarkY1-Train, *Profile Entities* achieves $MAP = 0.32$ whereas *Profile Terms* achieves $MAP = 0.27$. Similarly, on BenchmarkY2-Test, *Profile Entities* achieves $MAP = 0.39$ whereas *Profile Terms* achieves $MAP = 0.34$. Moreover, from the difficulty test for BenchmarkY1-Train in Figure 5.4b, we observe that the contribution of the profile entity features is always more than that of the profile term features. This shows that profile entities are more informative than profile terms.

**Conclusions.** With respect to **RQ4**, we may say that although the global context of the target entity can provide useful information for support passage retrieval, it is more useful to use the local context as it provides a query-specific background information on the target entity. Moreover, entities in local context contribute more to the retrieval performance than the terms.

# CHAPTER 6
# CONCLUSION

In this work, we address the problem of entity support passage retrieval. We present a novel method which identifies the query-relevant entity information in candidate support passages using the local context of the target entity. The local context is obtained from the query-relevant passages mentioning the target entity. Such local context is incorporated using a distribution over the frequently co-occurring entities with the target entity, and a distribution of frequent words in the context of the target entity. We show that our method achieves new state-of-the-art results on the task.

We propose a model called *entity prominence* which scores a candidate support passage for an entity in the context of a query. Our model uses the entities which occur frequently with in the local context of the target entity. The scoring of a candidate support passage is based on the intuition that a good support passage would mention many entities which frequently co-occur with the target entity. We show that our proposed method achieves new state-of-the-art results on the task.

We also explore the utility of entity salience for support passage retrieval. We use the salience of the target entity in the support passage to find good support passages for a given target entity. Our experiments show that although the usefulness of entity salience-based methods depends on quality of the underlying candidate passage ranking being used, salience can help improve retrieval effectiveness over the current state-of-the-art methods in the field.

To study the importance of the local entity context versus a global context, we treat we treat each Wikipedia page as an entity and use the information from the Wikipedia article

of the target entity as the global context. We find that entity information derived using the global context of the target entity are as good as those derived using the local context. However, the local context provides better entity information as they are query-relevant and query-dependent.

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.

## LIST OF REFERENCES


[1] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[2] Karen Sparck Jones. *A Statistical Interpretation of Term Specificity and Its Application in Retrieval*, page 132–142. Taylor Graham Publishing, GBR, 1988.

[3] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[4] 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.

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

[6] 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.

[7] 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.

[8] Jay Michael Ponte and W Bruce Croft. *A language modeling approach to information retrieval*. PhD thesis, University of Massachusetts at Amherst, 1998.

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

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

[11] 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.

[12] Laura Dietz, Michael Schuhmacher, and Simone Paolo Ponzetto. Queripidia: Query-specific wikipedia construction. *Proc. of AKBC-14*, 2014.

[13] 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.

[14] Ian Soboroff, Arjen P de Vries, and Nick Craswell. Overview of the trec 2006 enterprise track. In *Trec*, volume 6, pages 1–20, 2006.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] Ian Soboroff, Shudong Huang, and Donna Harman. Trec 2018 news track overview. In *TREC*, 2018.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Marco Ponza, Paolo Ferragina, and Francesco Piccinno. Swat: A system for detecting salient wikipedia entities in texts. *Computational Intelligence*, 04 2018.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

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

[46] 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.

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

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

[49] Xitong Liu and Hui Fang. Latent entity space: A novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] Joseph Rocchio. Relevance feedback in information retrieval. *The SMART Retrieval System–Experiments in Automatic Document Processing*, pages 313–323, 1971.

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

[63] Michael Bendersky, Donald Metzler, and W. Bruce Croft. Effective query formulation with multiple information sources. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 443–452, New York, NY, USA, 2012. Association for Computing Machinery.

[64] Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 154–161, New York, NY, USA, 2006. Association for Computing Machinery.

[65] Wouter Weerkamp, Krisztian Balog, and Maarten de Rijke. Exploiting external collections for query expansion. *ACM Trans. Web*, 6(4), November 2012.

[66] 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.

[67] 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.

[68] 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.