Entity-Based Retrieval

Research Thesis

Submitted In Partial Fulfillment of the

Requirements for the

Degree of Doctor of Philosophy

Hadas Raviv

Submitted to the Senate of the Technion - Israel Institute of Technology

Nisan, 5778, Haifa, March 2018

The Research Thesis was Done Under the Supervision of Associate Professor Oren Kurland in the Faculty of Industrial Engineering and Management.

THE GENEROUS FINANCIAL HELP OF THE TECHNION IS GRATEFULLY ACKNOWLEDGED.

List of publications:

- Hadas Raviv, Oren Kurland and David Carmel. "Document Retrieval Using Entity-Based Language Models". In Proceedings of 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 65-74, 2016 (Full paper).
- Hadas Raviv, Oren Kurland and David Carmel. "Query performance prediction for entity retrieval". In Proceedings of 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 10991102, 2014. (Short paper)
- Hadas Raviv, Oren Kurland and David Carmel. "The cluster hypothesis for entity oriented search". In Proceedings of 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 841844, 2013. (Short paper)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The abundance of digital information makes search essential in our everyday life. One of the most important tasks in the field of Information Retrieval (IR) is to effectively identify information that pertains to a user's information need, expressed by a search query.

In this work we study how entities, which are semantically meaningful units associated with rich semantic information, can be utilized for addressing users' information needs. We address two different tasks: entity retrieval and entity-based ad hoc document retrieval.

Entity retrieval is the task of ranking entities in a repository with respect to a user query. In this work we present the first study of the cluster hypothesis (cf. [144]) for entities: *"closely associated entities tend to be relevant to the same requests"*. We show that the hypothesis holds to a substantial extent for the task of entity retrieval. In addition, we suggest a novel cluster-based method for entity ranking which is shown to be highly effective. Finally, we explore the query performance prediction task for entity retrieval; that is, estimating retrieval effectiveness without having relevance judgments.

Ad hoc document retrieval is the classic IR task of ranking documents with respect to a query. Traditionally, this task is addressed by comparing term-based query and document representations. We present novel entity-based query and document representations which are based on language models. The models integrate entity and term information. We show that using these language models for retrieval significantly improves retrieval effectiveness with respect to using terms or entities alone, and with respect to a state-of-the-art term-proximity-based retrieval method.

Finally, we devise novel methods for inducing entity-based query models by utilizing inter-entity similarities. We evaluate the retrieval effectiveness of using these query models and demonstrate their considerable potential for estimating document relevance.

Chapter 1
Introduction

One of the major challenges in information retrieval (IR) is the need to effectively identify information that pertains to a user's information need, formulated by a query. Users expect a short and accurate response, especially when search is being performed using small mobile devices.

It has been recently observed that many users' information needs are centered around *entities*, which are semantically meaningful units of information. Different studies show that about 70% of Web search queries contain named entities such as people and locations [64, 97, 124]. The primary intent of almost 60% of these queries is associated with the entities they contain [124].

These findings have led to an increasing interest in utilizing entities for addressing users' information needs. On one hand, entities in a repository (e.g., Wikipedia or Freebase), can serve as "retrieval units" and be retrieved in response to a query instead of documents. The goal of a task named *entity retrieval* is: *"addressing information needs that are better answered by returning specific objects (entities) instead of just any type of documents"* [107]. The core challenge in addressing the entity retrieval task is estimating the relevance of a given *entity* to a query.

On the other hand, the semantic information associated with entities can be utilized for addressing a very fundamental task in information retrieval: *ad hoc document retrieval*. The goal is estimating the relevance of a given *document* in a corpus with respect to a user's query. Many methods utilizing bag-of-terms text representations were proposed for addressing the ad hoc document retrieval task (e.g., [123, 137, 136]); often, the text is represented by the counts of terms it contains and their corpus statistics. Such representations pose a serious limitation in terms of the ability to accurately estimate document relevance, as terms are often not effective enough for representing the text *meaning*. The assumption underlying the use of entities for document retrieval is that entity-based representations can bear semantic meaning, which would be helpful for the task of estimating document relevance.

The main research question we address in this work is:

How can entity-based information be utilized to address users' information needs expressed using queries?

We address this question by exploring the two tasks discussed above. First, we address the task of entity retrieval and propose new methods for entity ranking and for automatically estimating entity retrieval effectiveness, without having relevance judgments. Second, we propose novel types of query and document entity-based representations that are utilized for the task of ad hoc document retrieval. Finally, we propose additional novel methods for inducing entity-based query models, by utilizing a specific type of entity associated information: inter-entity similarities. We now turn to survey our contributions in each of these research directions.

1.1 Entity Retrieval

The task of entity retrieval has been addressed by several evaluation campaigns, each focused on a different type of entity-related information needs [13, 15, 16, 22, 39, 44, 45, 47, 149]. In Chapter 2, we survey the main campaigns that were proposed, as well as the important approaches suggested for addressing them.

In this thesis, we focus on a specific entity retrieval task: entity ranking in Wikipedia [44, 45, 47]. The goal of this task is ranking entities by their presumed relevance to a query in the English Wikipedia. It is assumed that each page in Wikipedia corresponds to an entity. The queries used in this task represent information needs that are aimed at finding entities of a defined type. Such information needs were found to be highly popular in Web search [124].

In Chapter 3, we explore the cluster hypothesis [144], a fundamental concept in ad hoc document retrieval, for the task of entity ranking in Wikipedia. The hypothesis we state is: *"closely associated entities tend to be relevant to the same requests"*. We use Voorhees' nearest-neighbor test [146] for testing the proposed hypothesis. Several inter-entity similarity measures are used. These measures utilize the categories associated with entities in their Wikipedia pages. We show that the hypothesis, as measured using the nearest-neighbor test, holds to a substantial extent for the task of entity retrieval.

Motivated by the findings about the cluster hypothesis for entity ranking we propose a novel cluster-based method for entity ranking in Chapter 3. We show that ranking entity clusters by the percentage of relevant entities they contain results in extremely effective retrieval of entities. This demonstrates the considerable potential of using clusters of similar entities for entity retrieval. In addition, we propose a method for ranking clusters of similar entities. Using this method to induce entity ranking results in an effective entity retrieval performance [127].

In Chapter 3 we also address the Query Performance Prediction (QPP) task for entity retrieval; that is, estimating retrieval effectiveness without having relevance judgments. First, we show how to adapt state-of-the-art query-performance predictors proposed for document retrieval to the entity retrieval domain. We use prediction methods that can be categorized into two groups [28]: pre-retrieval predictors directly utilize the query expression for assessing its "difficulty" [66]; post-retrieval predictors also analyze the *entity result list*: the list of entities most highly ranked in response to the query [28]. We show that predictors of both types can be successfully applied to the task of entity retrieval. Also, we show that different types of properties associated with entities (e.g., Wikipedia document, categories) can be successfully utilized for predicting retrieval performance.

Second, we present a novel predictor for entity retrieval that is based on the cluster hypothesis evaluated in Chapter 3. The suggested predictor utilizes retrieval scores of entity clusters to estimate retrieval effectiveness. An empirical evaluation shows that our suggested predictor can often outperform the most effective predictors we experimented with.

1.2 Entity-Based Ad Hoc Document Retrieval

Ad hoc document retrieval is the classic IR task of ranking documents with respect to a user query. Traditionally, this task is addressed by comparing query and documents representations. The underlying assumption is that the query-document similarity estimate is correlated with document relevance.

A commonly used query and document representation is referred to as "bag-of-terms" representation (e.g., [123, 136]). Texts are represented by counts of the different terms constituting them. Term occurrence statistics in the corpus are often utilized as well [123, 130, 132, 136].

Similarity estimates computed by comparing query and documents bag-of-terms representations are not always strongly correlated with document relevance, for two main reasons. First, mismatch between the vocabulary used in the query and that used in relevant documents can result in ineffective similarity estimates. This is a fundamental challenge of the search task named the "vocabulary mismatch problem" [83]. Second, many concepts in natural languages are represented by term sequences of varying lengths. These concepts are not well captured when using bag-of-terms representations. In Chapter 4 we survey previously proposed methods addressing these two challenges.

In Chapter 5, we propose novel query and document representations for document relevance estimation. Using these representations helps to address the challenges discussed above. To create these representations we use a highly effective technology, developed recently, which is named *entity linking* [38]. Entity linking tools mark terms or term sequences in a text as entities in an entity repository, and provide a confidence score which reflects the likelihood that a term sequence corresponds to an entity. Our proposed query and document representations are language models that utilize *markups* of entities in a text and terms it contains. The language models serve for retrieval in the language modeling framework. The main novelty of these language models is accounting, simultaneously, for (i) the uncertainty in entity linking reflected by the confidence score assigned to an entity markup and, (ii) the balance between using term-based and entity-based information.

In Chapter 5, we present an empirical evaluation which demonstrates the merits of using our entity-based language models for retrieval. We show that utilizing both term and entity information results in retrieval performance that significantly transcends that of using entities or terms alone. We also show that the retrieval performance attained by our suggested models is significantly better than that of a state-of-the-art term proximity method: the sequential dependence model (SDM) [111, 77]. The language models are also found to be effective for two additional retrieval paradigms: cluster-based document retrieval and query expansion.

Motivated by our empirical findings, in Chapter 6 we turn to address an additional challenge associated with estimating the similarity of documents to a query. In the ad hoc document retrieval task, a user information need is expressed by a

query which is usually short. More informative representation is often required for an effective query-document similarity estimation.

We suggest using our entity-based language models, together with additional entity associated information, for inducing query models that better reflect the underlying information need. Specifically, we explore the merits of using inter-entity similarity estimates for inducing entity-based query models. As a first step, we propose a novel cluster hypothesis (cf. [144]) for entities named the second-order cluster hypothesis: *closely associated entities tend to be relevant to the same requests*. Differently from the hypothesis stated in Chapter 2, this hypothesis is stated for the ad hoc document retrieval task. The underlying assumption is that the type of retrieved item (document) can be different from the type of the item for which the hypothesis is stated (entities). Therefore, we term our hypothesis "second-order". We suggest a novel method for estimating entity relevance with respect to a given query. These estimates, as well as various inter-entity similarity measures, are then utilized for testing the proposed second-order cluster hypothesis. Our empirical findings are that the hypothesis holds to a substantial extent for all inter-entity similarity measures we consider.

Next, we propose methods for inducing entity-based query models by utilizing inter-entity similarities. An empirical evaluation which demonstrates the merits of using these models for retrieval is presented in Chapter 6. Specifically, we show that retrieval methods utilizing our proposed query models are highly effective with respect to a few effective baselines. In addition, we perform oracle experiments which demonstrate the considerable potential of using clusters of similar entities to induce effective query models.

Chapter 2
Related Work - Entity Retrieval

The main goal of the entity retrieval task is to rank *entities*, instead of *documents*, with respect to a query by their presumed relevance to the information need that the query expresses. Several evaluation campaigns for entity retrieval, each focused on a different aspect of this task, have been proposed. For example, the goal of the TREC's expert finding task (2005-2008) was to rank employees in the enterprise by their expertise on a topic [39]. The goal of the INEX entity ranking track (2007-2009) was to retrieve entities that pertain to a query in the English Wikipedia [44, 45, 47]. The goal of the related entity finding task in TREC's entity track (2009-2011) was to rank Web entities with respect to a given entity by their relationships [13, 15, 16]. The Semantic Search Challenge (2010-2011) [22] was focused on searching entities over the Web of Data. Most recently, the task of ad hoc entity search over Wikipedia and the Web of Data, was suggested and explored in the INEX Linked Data Track (2012-2013) [149].

Entity retrieval is different from standard document retrieval in two main aspects. First and most important, entities are not well defined; for example, entities are identified differently in the entity retrieval tasks mentioned above. In the INEX entity ranking track, an entity is identified by a unique Wikipedia page ID, while in the TREC entity track, an entity is identified by a unique Web homepage. In the Semantic Search Challenge, an entity is identified by its URI in the Web of Data, whereas in the TREC expert finding task, experts are identified by their email addresses. In addition, entities are not always organized as units that can be retrieved in response to a query. In some of the datasets used for evaluating entity retrieval tasks, information about entities and their associated properties must be automatically extracted and analyzed by the search system to enable effective entity search.

Entity representation is directly related to the second challenge associated with entity search. Since entity representations are composed of both structured and unstructured data, traditional retrieval methods should be adjusted to enable full utilization of the entity associated information for retrieval. How to effectively exploit the entity associated information for retrieval is still an open challenge [9].

This chapter is structured as follows. In Section 2.1, we provide an extended background about entity retrieval. We elaborate on the INEX entity ranking task which we address in our work. In Section 2.2, we discuss entity representations in general as well as specifically in the INEX entity ranking task. In Section 2.3, we present entity ranking models which were proposed in the past and are relevant to our work. Finally, in Section 2.4, we describe the query performance prediction task [28] suggested for ad hoc document retrieval, which we address for entity retrieval in Chapter 3.

2.1 Entity Search Evaluation Campaigns

Pound et al. [124] suggested a high-level categorization of entity related queries according to several major search intents identified in a large search engine query log. They found that the most popular entity related queries are aimed at finding specific entities or entities of a particular type. Similar findings were reported by Lin et al. [97]. Associating queries with intents is important since different retrieval methods can be applied to address different information needs. Identifying and successfully classifying search intents of entity related queries is still an open challenge [31, 62, 161, 164].

Each of the different entity search tasks mentioned above focuses on a specific entity-related search intent. For example, the goal of one of the tasks in the Semantic Search Challenge is finding specific entities in the Web of Data [22]. Queries aimed at finding a specific entity are the most popular in the Web. Still, challenges such as entity disambiguation are involved in addressing such information needs.

Tasks centered on finding entities of a specific type include the entity ranking task in the INEX entity ranking track [44, 45, 47] and the expert finding task at the TREC Enterprise Track [39]. In these tasks, a description which relates to a Wikipedia category or to a specific type of expertise, respectively, is provided in natural language. Entities of the described type should be ranked with respect to the query. In Chapter 3, we suggest retrieval methods for addressing the INEX entity ranking task with the objective of ranking entities of a particular type. We therefore elaborate on the INEX entity ranking track in the following.

The INEX entity ranking track took place during 2007-2009 and ran two tasks [44, 45, 47]. The entity ranking task's goal was to retrieve entities most relevant to a given topic from the Wikipedia collection. The topic included a title which usually served as a query or as a part of a query. In addition, candidate entities were restricted to be items having their own Wikipedia article. The recommended types of entities to retrieve (the entity target types) were defined in the search topic by a list of Wikipedia categories. Returning entities having the exact category provided by the topic was not mandatory. However, experimental results have shown that utilizing this information is crucial for retrieval effectiveness.

An additional task explored in the INEX entity ranking track was the entity list completion task. The goal was ranking Wikipedia entities with respect to a topic that included example entities, along with a textual description of the information need.

The same INEX topics were used for both tasks. Figure 2.1 shows an example topic, used in the 2007 track. The 2007 and 2008 tracks utilized the English Wikipedia dataset from 2006, composed of $659,388$ documents. The 2009 track used the English Wikipedia from 2008, composed of $2,666,190$ documents. In this work we focus on the entity ranking task.

<inex topic topic id="67">
<title>Ferris and observation wheels< /title>
<entities>
<entity id="30372" >London Eye< /entity>
<entity id="490289" >Roue De Paris< /entity>
<entity id="2669944" >Singapore Flyer< /entity>
< /entities>
<categories>
<category>ferris wheels< /category>
< /categories>
<description> Find all the Ferris and Observation wheels in the world
< /description>
<narrative> I have been to the "*RouedeParis*" last Sunday and enjoyed it. I would like to know which other wheels exist or existed in the world, to find out the highest and what buildings you can see from each. < /narrative>
< /inex topic>

Figure 2.1: Sample topic of the INEX entity ranking and entity list completion tasks.

2.2 Entity Representation

Entity representations are created using a wide range of methods, depending on the dataset being used. In some of the datsets, entity related information is naturally organized within the collection [44, 45, 47, 65]. For example, RDF triplets in the Web of Data are composed of URIs which uniquely identify entities [65]. In Wikipedia, entities are identified by their page, usually composed of various information types that can be utilized as the entity properties (e.g., Wikipedia categories) [44, 45, 47]. In other datasets, for example the ClueWeb09 collection, an unstructured crawl of the Web used in the TREC entity ranking track [15], entity related information is not well defined. Advanced natural language processing methods must be applied to generate informative entity representations [150, 151].

The underlying assumption of using Wikipedia for entity retrieval is that each page in the collection corresponds to an entity [44, 45, 47]. Systems performing retrieval of entities from Wikipedia use all or few of the related page properties for creating an entity representation [12, 81, 127]. A commonly used page property is the text it contains, referred to as the *entity document*. The list of Wikipedia categories associated with the entity page are referred to as the *entity type*. The page title is the *entity name*. Additional names are sometimes extracted from Wikipedia's redirect pages. Finally, incoming and outgoing links define the *entity relations*. The different entity properties are usually stored separately for retrieval purposes [81, 145].

2.3 Entity Retrieval Models

In this section we describe retrieval approaches that were suggested for addressing various entity retrieval tasks and which are the most related to our work.

2.3.1 Generative models for entity ranking

Generative models are popular in the context of entity retrieval due to their good empirical performance and their sound mathematical foundations. A general probabilistic framework for entity retrieval is developed by applying the *probability ranking principle* [131] to the task of entity retrieval. Specifically, an entity e is ranked with respect to the query q according to the probability $p(R = 1|q, e)$; R is a random variable denoting relevance, value of 1 indicates relevance.

Fang and Zhai [54] showed that using the probability ranking principle, two families of generative models for entity ranking can be induced. Methods utilizing one of these two approaches were proposed for addressing various entity ranking tasks [10, 12, 14, 21, 27, 43, 59, 104, 118, 174, 178]. *Candidate generation models* estimate the probability $p(e|q)$, that an entity e is generated from the query q. *Query generation models* estimate the probability $p(q|e)$, that the query q is generated from an entity e.

Balog et al. [10] have formally defined two principled ways of estimating the query generation probability, $p(q|e)$, for finding experts in an enterprise. The first approach, referred to as *Model 1*, utilizes textual representations induced for each entity in the collection using some representation induction method. Using this representation, language model-based retrieval is applied for estimating the query generation probability. The second method, referred to as *Model 2*, utilizes the collection documents for estimating the query generation probability. First, documents that are presumably relevant to the query are retrieved. Then, candidate entities associated with the documents are ranked according to their aggregated association level.

Many variations of these two models have been proposed for addressing different entity related tasks. The variations include using different types of entity representations [21, 118], various document retrieval models [32, 43, 46, 59, 102, 165], different measures for estimating the entity document association [11, 59, 122], incorporation of term proximity information [11, 122] and more.

Incorporating structure into Model 1 has been consistently shown to improve entity retrieval effectiveness. For example, the most successful retrieval methods over the Web of Data [21, 65, 118, 24] utilized fielded retrieval models, such as BM25F [130], mixture language models (MLM) [119] and their extensions [24, 118]. For addressing the task of entity ranking in Wikipedia, most successful retrieval methods utilized complex query and entity representations [12, 79]. For example, Balog et al. [12] and Jiang et al. [79] suggested methods for estimating the entity relevance by separately comparing two different query and entity representations. One representation utilizes terms in the topic's title and in the entity document.

The second representation utilizes terms constituting the entity associated categories and the topic's categories.

We do not directly use generative models in our work. However, we are inspired by the query generation models described above, and specifically by Model 1 [10]. We use various information types associated with a Wikipedia entity to estimate its relevance with respect to the query and to estimate retrieval effectiveness in the task of query performance prediction for entity retrieval [28].

2.3.2 Discriminative models for entity ranking

Discriminative models directly estimate the probability $P(R = 1|q, e)$, i.e., the probability that an entity e is relevant to a query q, by integrating feature functions. Such models were applied for a variety of entity search tasks [29, 36, 55, 81, 98, 117, 141, 145, 154, 177].

An important group of discriminative models utilized for entity ranking is learning to rank (L2R) based models [98]. The suggested models [14, 29, 36, 55, 98, 117, 154] utilize features measuring the similarity between various entity properties (e.g., entity type, entity name) and the query. L2R-based models have been shown to consistently outperform generative entity ranking models [14, 55].

The relevance of an entity in Wikipedia with respect to a query was estimated by a few proposed methods [81, 141, 145, 177] using a set of heuristic features utilizing various entity properties. For example, Vercoustre et al. [145] and Kaptein and Kamps [81] used three scores for ranking a candidate entity that are based on the entity document, type and associated links. Entity type was shown to be an important factor in estimating the relevance of an entity with respect to the query in the INEX entity ranking track.

Dalton et al. [43] and Zhiltsov et al. [175] used the Markov Random Field (MRF) framework [111] for modeling term proximity when retrieving entities from the Web of Data (WOD). Similarly to the case in standard ad hoc document retrieval [111], considering term proximity for entity ranking was shown to improve retrieval effectiveness in comparison to a simple unigram-based ranking [21].

In Chapter 3, we present a cluster-based retrieval method for addressing the entity ranking task. In addition, we explore the query performance prediction task for entity retrieval. Both these works rely on a discriminative entity retrieval model that ranks entities by aggregating different relevance estimates for entities. This model was published in [127] and is inspired by the retrieval methods described above.

2.3.3 Cluster-based models for entity ranking

The *Cluster Hypothesis* is an important principle in ad hoc document retrieval: *"closely associated documents tend to be relevant to the same requests"* [144]. Several tests of the cluster hypothesis for document retrieval were proposed [52, 78, 139, 146]. One of them is Voorhees' nearest-neighbor test [146], which we use in our

work. The nearest neighbors test considers the percentage of relevant documents among the k nearest neighbors of a relevant document. This number is averaged over relevant documents to determine the extent to which the cluster hypothesis holds. Voorhees' nearest neighbor test relies on inter-document similarities. Recently, the test was applied with various inter-document similarity measures [126].

In Section 3.1 we present the first study of the cluster hypothesis for entity ranking. We show that this hypothesis holds to a substantial extent for several inter-entity similarity measures. We measure similarity between two entities based on the similarity between sets of categories associated with their corresponding entity pages, with several variations.

Clusters can potentially improve ad hoc retrieval performance in one of two principal ways. First, they can be utilized for document selection [87, 100, 125, 143]: given a query q and a list of document clusters Cl (created offline or given a specific query), clusters are ranked by calculating a query-cluster similarity score. The cluster ranking is then transformed to document ranking by, for example, replacing each cluster with its constituent documents. Second, clusters can be used for enriching ("smoothing") document representations by incorporating information induced from similar documents [89]. Kurland and Lee [89] showed that cluster-based smoothing and cluster ranking are complementary for document retrieval.

In Section 3.1 we demonstrate the retrieval effectiveness merits of using clusters of similar entities for entity ranking. There are very few works on using cluster-based methods for entity ranking. In Cao et al.'s [26] and Yao et al. [67] works on expert search, the retrieval score assigned to an entity was smoothed with the retrieval score assigned to a cluster containing the entity. In contrast to these works, we explore a retrieval paradigm that ranks entity clusters and transforms the ranking to entity ranking.

Methods based on cluster ranking were also applied for ranking entities in the Web of Data. Ciglan et al. [33] used groups of semantically related entities, to re-rank a list of initially retrieved entities. The relevance of each set to the query was estimated and its members were ranked accordingly. In contrast to this work, we re-rank entities in Wikipedia. Due to the different collection characteristics, we use different methods for inducing the entity clusters and for ranking entity clusters with respect to the query.

Liang and de Rijke [96] proposed an extension of the expert finding task named "the group finding task". The task goal is ranking knowledgeable groups, i.e., entity clusters, in an enterprise corpora with respect to a given query. Several ranking methods were proposed using the language modeling framework. In contrast, our goal is ranking *individual entities* with respect to the query. We use cluster ranking as an intermediate step.

Overall, the findings we present for the entity ranking task echo those reported for document retrieval. Namely, we show that the cluster hypothesis holds to a substantial extent [146] and demonstrate the (potential) merits of using cluster ranking for entity retrieval [68, 87, 101, 143].

2.4 Query Performance Prediction

Query-Performance Prediction (QPP) methods proposed for document retrieval can be categorized into two groups [28]. Pre-retrieval predictors analyze the query using corpus-based term statistics [66]. Post-retrieval predictors also analyze the result list of top-retrieved documents [28]. We adapt the most effective of these predictors to the entity retrieval task. The predictors we use are detailed in Section 3.2.

To the best of our knowledge, there is a single report of work on QPP for entity retrieval [105]. The entity-list completion task was addressed, where examples of relevant entities are provided. The most effective predictors used the description and narrative of the (INEX) topic as well as information induced from the example entities. In contrast, we address the entity ranking task, and the predictors we study do not use entity feedback (i.e., examples) nor the topic's narrative and description. We show that post-retrieval predictors outperform pre-retrieval predictors, which was not the case in this work [105], which did not adapt state-of-the-art predictors proposed for document retrieval.

Chapter 3
Entity Retrieval Models

In this chapter we address the task of entity retrieval: ranking entities with respect to a user query [107]. In Section 3.1, we explore the cluster hypothesis for entity retrieval and propose a novel cluster-based method for entity ranking. In Section 3.2, we explore the query performance prediction task (QPP) for entity retrieval; that is, estimating retrieval effectiveness without having relevance judgments.

3.1 The Cluster Hypothesis for Entity Retrieval

The entity retrieval task is different than the standard ad hoc document retrieval task as entities are somewhat more complex than (flat) documents. That is, entities are characterized by different properties such as name, type (e.g., place or person), and potentially, an associated document (e.g., a homepage or a Wikipedia page). Despite the fundamental difference between the two tasks, in this section we set as a goal to study whether an important principle in ad hoc document retrieval also holds for the entity retrieval task; namely, the *cluster hypothesis* [144]. We present the first study of the cluster hypothesis for entity retrieval, where the hypothesis is that "closely associated *entities* tend to be relevant to the same requests".

We use several inter-entity similarity measures to quantify the association between entities, which is a key point in the hypothesis. These measures are based on the entity type which is a highly important source of information [127, 82]. We then show that the cluster hypothesis, tested using Voorhees' nearest neighbor test [146], can hold to a substantial extent for entity retrieval for several of the similarity measures.

Motivated by the findings about the cluster hypothesis, we explore the merits of using clusters of similar entities for entity ranking. We show that ranking entity clusters by the percentage of relevant entities that they contain can be used to produce *extremely* effective entity ranking. We also demonstrate the effectiveness of using cluster ranking techniques that are based on estimating the percentage of relevant entities in the clusters for entity ranking.

Our main contributions are three fold: (i) showing that for several inter-entity similarity measures the cluster hypothesis holds for entity retrieval to a substantial extent as determined by the nearest neighbor test; (ii) demonstrating the considerable potential of using clusters of similar entities for entity retrieval; and, (iii) showing that using simple cluster ranking methods can help to improve retrieval performance with respect to that of an effective initial search.

3.1.1 The cluster hypothesis

Our first goal is to explore the extent to which the cluster hypothesis holds for entity retrieval. To this end, we use the nearest neighbor test [146]. Let $L_q^{[n]}$ be

the list of n entities that are the highest ranked by an initial search performed in response to query q. For each relevant entity in $L_q^{[n]}$, we record the percentage of relevant entities among its K nearest neighbors in $L_q^{[n]}$. The nearest neighbors are determined using one of the inter-entity similarity measures specified in Section 3.2.3.1. The test result is the average of the recorded percentages over all relevant entities in $L_q^{[n]}$, averaged over all test queries.

Some of the inter-entity similarity measures assign discrete values including 0. Hence, for some relevant entities there could be less than K neighbors as we do not consider neighbors with a 0 similarity value. In addition, a relevant entity might be assigned with more than K nearest neighbors due to ties in the similarity measure. That is, we keep collecting all entities having the same similarity value as that of the last one in the K neighbors list.

3.1.2 Cluster-based entity ranking

Our second goal is studying the potential merits of using entity clusters to induce entity ranking. We re-rank the initial entity list $L_q^{[n]}$ using a cluster-based paradigm which is very common in work on document retrieval [101]. Let $Cl(L_q^{[n]})$ be the set of clusters created from $L_q^{[n]}$ using *some* clustering method. The inter-entity similarity measures used for creating clusters are those used for testing the cluster hypothesis. (See Section 3.2.3.1 for further technical details.) The clusters in $Cl(L_q^{[n]})$ are ranked by the presumed percentage of relevant entities that they contain. Below we describe two cluster ranking methods. Then, each cluster is replaced with its constituent entities while omitting repeats. Within cluster entity ranking is based on the initial entity retrieval scores which were used to create the list $L_q^{[n]}$.

The **MeanScore** cluster ranking method scores cluster c by the mean retrieval score of its constituent entities: $\frac{1}{|c|}\sum_{e\in c} S_{init}(e;q)$; $S_{init}(e;q)$ is the initial retrieval score of entity e; $|c|$ is the number of entities in c.

When $S_{init}(e;q)$ is a rank equivalent estimate to that of $\log(Pr(q,e))$ [127], the cluster score assigned by MeanScore is rank equivalent to the geometric mean of the joint query-entity probabilities' estimates in the cluster. Using a geometric-mean-based representation for document clusters was shown to be highly effective for ranking document clusters [101].

The regularized mean score method, **RegMeanScore** in short, which is novel to this study, smoothes c's score: $\frac{\sum_{e\in c} S_{init}(e;q)+\frac{1}{n}\sum_{e\in L_q^{[n]}} S_{init}(e;q)}{|c|+1}$. The cluster score is the mean retrieval score of a cluster composed of c's entities and an additional "pseudo" entity whose score is the mean score in the initial list. This method helps to address, among others, cluster-size bias issues.

Data set	WP year	Collection size	#Documents in collection	Train topics	Test topics
2007	2006	4.4 GB	659,388	28	46
2008				74	35
2009	2008	50.7 GB	2,666,190	-	55

Table 3.1: INEX entity ranking datasets.

3.1.3 Evaluation

3.1.3.1 Experimental setup

We conducted experiments with the datasets of the INEX *entity ranking track* of 2007 [44], 2008 [47], and 2009 [45]. Table 3.1 provides a summary of the datasets. The tracks for 2007 and 2008 used the English Wikipedia dataset from 2006, while the 2009 track used the English Wikipedia from 2008. The set of test topics for 2007 is composed of 21 topics that were derived from the ad hoc 2007 assessments, and additional 25 topics that were created by the participants specifically for the track. In 2008, 35 topics were created and used for testing. The topics used for testing in 2009 were 55 topics out of the 60 test topics used in 2007 and 2008.

We used Lucene (http://lucene.apache.org/core/) for experiments. The data was pre-processed using Lucene, including tokenization, stopword removal, and Porter stemming.

Inter-entity similarity measures The inter-entity similarity measures that we use utilize Wikipedia categories. Specifically, the categories associated with the Wikipedia page of the entity, henceforth referred to as its category set, serve as the entity type.

The **Tree** similarity between two entities e_1 and e_2 is $\exp(-\alpha d(e_1, e_2))$ where $d(e_1, e_2)$ is the minimum distance over Wikipedia's categories graph between a category in e_1's category set and a category in e_2's category set; α is a decay constant determined as in [127].

The **SharedCat** measure is the cosine similarity between the binary vectors representing two entities. An entity vector is defined over the categories space. An entry in the vector is 1 if the corresponding category is associated with the entity and 0 otherwise. Thus, SharedCat measures the (normalized) number of categories shared by the two entities [141].

The **CE** measure is based on measuring the language-model-based similarity between the documents associated with the category sets of two entities [82]. More specifically, each category is represented in this case by the text that results from concatenating all Wikipedia pages associated with the category. The similarity between the texts x and y that represent two categories is $\exp(-CE(p_x^{[0]}(\cdot)||p_y^{[\mu]}(\cdot)))$; CE is the cross entropy measure; $p_z^{[\mu]}(\cdot)$ is the Dirichlet-smoothed unigram language model induced from z with the smoothing parameter μ (=1000). The CE similarity between two entities is defined as the maximal similarity, over all pairs of categories,

one in the first entity's category set and the other in the second entity's category set, of the texts representing the categories.

Finally, the **ESA** (Explicit Semantic Analysis) [61] similarity measure is the cosine between two vectors, each represents the category set of an entity. The vectors representing the category sets are defined over the entities space. The value of an entry in the vector is the number of the categories in the given category set that are associated with the corresponding entity. Using ESA to measure inter-entity similarity is novel to this study.

Three different initially retrieved entity lists, $L_q^{[n]}$, are used for both the cluster hypothesis test and cluster-based ranking. The lists are created in response to the query using highly effective entity retrieval methods [127]. The first list, L_{Doc}, is created by representing an entity with its Wikipedia document (page). The documents are ranked in response to the query using the standard language-model-based approach with Dirichlet-smoothed unigram language models and the cross entropy similarity measure. The second list, $L_{Doc;Type}$, is created by scoring entities with an interpolation of two scores. The first is that used to create the list L_{Doc}. The second is the similarity between the category set of the entity and the query target type (the set of categories that are relevant to the query, as defined by INEX topics). The Tree estimate described above is used for measuring similarity between the two category sets. The third list, $L_{Doc;Type;Name}$, is created by scoring an entity with an interpolation of the score used to create $L_{Doc;Type}$, and an estimate for the proximity-based association [127] between the query terms and the entity name (i.e., the title of its Wikipedia page) in the corpus. We employ the same train-test approach as in [127] to set the free-parameter values of the ranking methods used to create the initial lists. The number of entities in each initial list $L_q^{[n]}$ is $n = 50$.

We use a simple nearest neighbor clustering method to cluster entities in the initial list $L_q^{[n]}$. Specifically, each entity in $L_q^{[n]}$ and the K $(= 5)$ entities in $L_q^{[n]}$ that are the most similar to it, according to the inter-entity similarity measures described above, form a cluster. Using such small overlapping clusters was shown to be highly effective for cluster-based document retrieval [87, 89, 101]. We note that not all clusters necessarily contain $K + 1$ documents due to the reasons specified in Section 3.1.1. For consistency, we also use $K = 5$ in the cluster hypothesis test.

Following the INEX guidelines, the evaluation metric for INEX 2007 is mean average precision (MAP) while that for INEX 2008 and 2009 is infAP. We also report the precision of the top 5 entities (p@5). Statistically significant differences of retrieval performance are determined using the two tailed paired t-test with a 95% confidence level.

3.1.3.2 Experimental results

The cluster hypothesis Table 3.2 presents the results of the nearest neighbor cluster hypothesis test that was described in Section 3.1.1. The test is performed on the different initially retrieved entity lists using the various inter-entity similarity measures. We see that the average percentage of relevant entities among the

nearest neighbors of a relevant entity ranges between 30% and 53% across the various experimental settings. We also found out that, on average, the percentage of relevant entities in a list is often lower than 25% and can be as low as 10%. Thus, due to the relatively high percentage of relevant entities among the nearest neighbors of relevant entities, we can conclude that the cluster hypothesis holds to a substantial extent, according to the nearest neighbor test, with various inter-entity similarity measures.

Table 3.2 also shows that for most of the data sets and similarity measures the test results for the $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ lists are higher than for L_{Doc}. This finding is not surprising as $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ were created using entity-query similarity measures that account for category information, while the similarity measure used to create L_{Doc} does not use this information. The highest test results are obtained for the SharedCat similarity measure which, as noted above, measures the (normalized) number of shared categories between two entities.

Similarity measure	Initial list	2007	2008	2009
Tree	L_{Doc}	30.0	32.0	42.7
	$L_{Doc;Type}$	29.8	35.5	44.9
	$L_{Doc;Type;Name}$	32.7	37.7	44.9
SharedCat	L_{Doc}	35.7	41.0	45.4
	$L_{Doc;Type}$	33.5	45.5	52.2
	$L_{Doc;Type;Name}$	37.9	44.3	52.7
CE	L_{Doc}	33.4	36.2	46.0
	$L_{Doc;Type}$	34.5	38.6	50.3
	$L_{Doc;Type;Name}$	37.5	41.7	49.7
ESA	L_{Doc}	34.3	36.2	46.2
	$L_{Doc;Type}$	33.7	41.0	49.5
	$L_{Doc;Type;Name}$	37.3	39.1	49.0

Table 3.2: The cluster hypothesis test: the average percentage of relevant entities among the 5 nearest neighbors of a relevant entity.

Cluster-based entity ranking Table 3.3 presents the results of employing cluster-based entity re-ranking, as described in Section 3.1.2, upon the three initial entity lists. The various inter-entity similarity measures are used for creating the clusters. 'Initial' refers to the initial ranking of a list. 'Oracle' is the ranking of entities that results from employing the cluster-based re-ranking paradigm described in Section 3.1.2; the clusters are ranked by the *true* percentage of relevant entities that they contain.

The high performance numbers for Oracle, which are substantially and statistically significantly better than those for Initial, attest to the existence of clusters that contain a very high percentage of relevant entities. More generally, these numbers attest to the incredible potential of employing effective cluster ranking methods to rank entities.

RegMeanScore, which outperforms MeanScore due to the regularization discussed in Section 3.1.2, is in quite a few cases more effective than Initial; specif-

ically, using the Tree and SharedCat inter-entity similarity measures. While the improvements for L_{Doc} are often statistically significant, this is not the case for $L_{Doc;Type}$ and $L_{Doc;Type;Name}$. Naturally, the more effective the initial ranking (Initial), the more challenging the re-ranking task. Yet, the very high Oracle numbers for $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ imply that effective cluster ranking methods can yield performance that is much better than that of the initial ranking. Finally, for both $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ the best performance is in most cases attained by using RegMeanScore.

$\mathbf{L_{Doc}}$						
	2007		2008		2009	
	MAP	p@5	infAP	p@5	infAP	p@5
Initial	20.2	26.1	12.6	19.4	19.1	35.3
Tree						
Oracle	31.8^{i}	51.3^{i}	22.3^{i}	49.7^{i}	25.5^{i}	68.0^{i}
MeanScore	21.6	26.1	13.3	17.1	20.2^{i}	36.7
RegMeanScore	21.6	26.0	13.4	18.9	$\mathbf{20.2^{i}}$	36.7
SharedCat						
Oracle	36.2^{i}	60.0^{i}	26.8^{i}	66.3^{i}	30.5^{i}	83.3^{i}
MeanScore	22.5	23.9	12.6	18.9	19.7	37.1
RegMeanScore	$\mathbf{23.1}^{i}$	$\mathbf{27.8}^{i}$	13.4^{i}	$\mathbf{20.6}^{i}$	19.9	**38.5**
CE						
Oracle	32.3^{i}	53.5^{i}	23.3^{i}	53.7^{i}	28.0^{i}	74.2^{i}
MeanScore	22.6	27.4	13.5	20.6	19.1	36.7
RegMeanScore	22.0	26.5	**13.5**	20.6	19.1	36.7
ESA						
Oracle	33.7^{i}	57.0^{i}	26.0^{i}	60.6^{i}	28.8^{i}	80.0^{i}
MeanScore	20.9	22.6	12.8	14.9	19.2	35.6
RegMeanScore	21.9	23.9	12.9	14.9	19.2	35.3

$\mathbf{L_{Doc;Type}}$						
	2007		2008		2009	
	MAP	p@5	infAP	p@5	infAP	p@5
Initial	30.8	37.4	28.2	**44.0**	**23.8**	43.6
Tree						
Oracle	37.7^{i}	58.7^{i}	32.5^{i}	50.9^{i}	29.5^{i}	70.2^{i}
MeanScore	31.6	40.0	27.7	37.7	23.6	39.6
RegMeanScore	**31.6**	**40.0**	28.1	40.6	23.4	39.3
SharedCat						
Oracle	43.8^{i}	65.7^{i}	38.3^{i}	65.1^{i}	34.1^{i}	87.6^{i}
MeanScore	30.8	36.1	28.7	42.3	23.2	44.0
RegMeanScore	31.1	37.0	**28.9**	42.3	23.6	**45.1**
CE						
Oracle	39.0^{i}	60.0^{i}	34.3^{i}	58.3^{i}	31.3^{i}	75.6^{i}
MeanScore	31.3	38.3	28.5	40.6	23.7	42.2
RegMeanScore	31.0	37.4	28.7	40.0	23.7	42.2
ESA						
Oracle	42.1^{i}	64.3^{i}	37.7^{i}	66.9^{i}	32.9^{i}	83.6^{i}
MeanScore	28.6	34.3	28.4	42.3	22.8	42.5
RegMeanScore	29.1	34.8	28.9	44.0	22.7	41.5

$\mathbf{L_{Doc;Type;Name}}$						
	2007		2008		2009	
	MAP	p@5	infAP	p@5	infAP	p@5
Initial	33.3	40.4	35.4	**46.9**	24.4	**44.0**
Tree						
Oracle	39.6^{i}	57.4^{i}	42.3^{i}	62.3^{i}	30.3^{i}	72.0^{i}
MeanScore	34.1	43.5	35.7	42.3	24.7	42.9
RegMeanScore	**34.0**	**42.6**	**35.7**	43.4	24.6	42.2
SharedCat						
Oracle	47.4^{i}	70.4^{i}	47.8^{i}	74.9^{i}	34.3^{i}	$87.^{i}3$
MeanScore	32.9	38.7	33.5	42.9	24.6	41.5
RegMeanScore	33.1	39.1	34.8	44.0	**24.9**	42.9
CE						
Oracle	40.7^{i}	59.1^{i}	43.1^{i}	63.4^{i}	31.6^{i}	76.4^{i}
MeanScore	33.6	38.7	34.5	45.1	24.6	43.3
RegMeanScore	33.6	38.7	34.5	45.1	24.8	43.6
ESA						
Oracle	44.7^{i}	66.5^{i}	45.7^{i}	70.3^{i}	32.9^{i}	81.5^{i}
MeanScore	33.9	41.3	33.7	43.4	23.0	35.3
RegMeanScore	34.0	42.2	34.2	44.6	23.3	35.6

Table 3.3: Retrieval performance. The best result in a column (excluding that of Oracle) per an initial list is boldfaced. 'i' marks statistically significant differences with Initial.

3.2 Query Performance Prediction for Entity Retrieval

In this section we address the *query-performance-prediction* (QPP) task for entity retrieval. The goal is to estimate, without relevance judgements, the effectiveness of retrieval performed in response to a query. While there is a large body of work on QPP for document retrieval [28], there has been very little work on QPP for entity retrieval [105]. Yet, the same motivation that triggered the development of predictors for document retrieval holds for entity retrieval. For example, alerts for ineffective retrieval can direct users to better formulate their queries.

We present a study of adapting state-of-the-art query-performance predictors, proposed for document retrieval, to the entity retrieval domain. In addition, we present a novel query-performance predictor for entity retrieval. The predictor relies on retrieval scores of clustered entities, following our study of the cluster hypothesis for entity retrieval in Section 3.1. Evaluation performed with the INEX entity ranking track collections shows that our novel predictor can often outperform the most effective predictors we experimented with.

3.2.1 QPP for entity retrieval

Our focus is on predicting retrieval performance for queries whose goal is finding entities of a particular type or class [124]. We use the datasets of the INEX *entity ranking track* [44, 45]. Each entity in the corpus is represented as a Wikipedia page associated with a set of categories which serve as the entity's type. The entity ranking task queries are composed of a short keyword-based title and a set of categories representing the query's target type. Entities relevant to the query are expected to be associated with categories in the query's target type, or with categories that are "close" to those in the target type in the Wikipedia category graph.

Most entity retrieval methods utilize several properties of entities [141, 12, 129]. Typical properties are the document associated with the entity (the Wikipedia page in our case), the entity type (the set of categories associated with the entity in our case), the entity name (the Wikipedia page title), etc. Accordingly, we study prediction methods that use information induced from two properties which were found to be highly effective for retrieval [12, 129]; namely, the document associated with the entity and the entity type.

Specifically, the prediction methods that we present use three entity representations. The first is **doc**, under which an entity is represented by its associated document. The second representation, **type**, is the bag of terms that appear in the names of the categories that constitute the entity type. Unless otherwise stated, whenever the *doc* and *type* representations are used, we use the set of terms in the query title and the set of terms in the names of the categories which constitute the query target type, respectively. The third entity representation, **score**, is the retrieval score assigned to the entity. The score can rely on either (or both) properties of the entity (its associated document and its type).

Below we present query-performance prediction approaches, denoted $\mathcal{P}$, which utilize the entity representations. We use $\mathcal{P}_{rep=r}$, where $r \in \{doc, type, score\}$ is the entity representation used by $\mathcal{P}$, to denote the resultant prediction methods.

Some of the predictors we explore utilize inter-entity similarity measures. The first measure, referred to as $Sim(=, doc)$, is the language-model-based similarity between the (Wikipedia) documents associated with the entities. The similarity between documents x and y is $\exp(-CE(p_x^{[0]}(\cdot)||p_y^{[1000]}(\cdot)))$; CE is the cross entropy measure; $p_z^{[\mu]}(\cdot)$ is the Dirichlet-smoothed unigram language model induced from z with the smoothing parameter μ. The second inter-entity similarity measure is based on the entity type: $Sim(=, type)$. The measure is the cosine similarity between the binary vectors that represent two entities in the category space. An entry in the vector is 1 if the corresponding category is associated with the entity and 0 otherwise.

To integrate a predictor which uses the *doc* entity representation (inter-entity similarity measure) with a predictor which uses the *type* representation (inter-entity similarity measure) we multiply the prediction values and denote the integration as $rep = doc \wedge type$ ($Sim(=, doc \wedge type)$).

3.2.2 Prediction approaches

3.2.2.1 Pre-retrieval predictors

Pre-retrieval prediction methods analyze the query using corpus-based term statistics prior to retrieval. We adapt two highly effective pre-retrieval methods from document retrieval to the entity retrieval setting.

The first type of predictors is based on analyzing the inverse document frequency (IDF) values of the set of terms in the query title; the *doc* entity representation is used. The resultant predictors are named $AIDF_{rep=doc}$ (cf., [40, 66]), where $A \in \{avg, sum, max\}$ is the aggregation type (average, summation, maximization) of the terms' IDF values.

We also use the IDF values of the set of terms that appear in the names of the categories that constitute the query target type. The *type* entity representation is used yielding the $AIDF_{rep=type}$ predictor.

The predictors just described quantify the discriminative power of the query by analyzing the IDF values of either its title or target type terms. Along the same lines, we study the $AVarTF.IDF_{rep=doc}$ predictor (cf. [173]) which measures for each query title term the variance of its *tf-idf* values across all the entity documents that contain it[1].

[1]Experiments showed that using the $VarTF.IDF$ predictors with the *type* entity representation yields poor prediction quality. Actual numbers are omitted as they convey no additional insight.

3.2.2.2 Post-retrieval predictors

We now describe post-retrieval predictors that analyze the n most highly ranked entities in a result list retrieved by an entity retrieval method; n is a free parameter.

Clarity The *Clarity* prediction method [40], proposed for document retrieval, is based on the premise that the more focused the result list with respect to the corpus the more effective the retrieval. Specifically, the KL divergence between a relevance language model [93] induced from the result list and a language model induced from the corpus is used to measure focus. For the entity retrieval task, we use the *doc* entity representation for *Clarity* computation. The resultant $Clarity_{rep=doc}$ predictor is the analogue of the *Clarity* predictor used for document retrieval [40]. Alternatively, the focus of the entity result list can be measured using the *type* entity representation, yielding the $Clarity_{rep=type}$ predictor. Particularly, a relevance language model induced from the bags of terms that represent the entity types is used.

QF Query feedback (QF) [176] is based on measuring the robustness of the result list. Specifically, a relevance model is constructed from the original result list and is used to retrieve a second list from the corpus. The overlap between the two lists, measured by the number of documents which are at the l_{qf} highest ranks of both lists, is used for prediction; l_{qf} is a free parameter. Higher prediction value presumably attests to improved robustness of the result list, and therefore to increased retrieval effectiveness. For the entity retrieval task, we simply use the *doc* entity representation for QF computation. The resultant $QF_{rep=doc}$ predictor is the analogue of that used for document retrieval.

WIG **and** *NQC* The *WIG* [176] and *NQC* [138] methods measure the mean and standard deviation, respectively, of document retrieval scores in the result list. To apply *WIG* and *NQC* for the entity retrieval task, we use the *score* entity representation; i.e, the retrieval scores of entities in the result list are utilized. The resultant predictors are $NQC_{rep=score}$ and $WIG_{rep=score}$, respectively.[2]

Cohesion It was suggested that a cohesive document result list indicates effective retrieval [28]. We measure the cohesion of the entity result list by the average similarity between two entities in the list using the *doc* and *type* inter-entity similarity measures. The resultant predictors are denoted $Cohesion_{Sim(=,doc)}$ and $Cohesion_{Sim(=,type)}$, respectively.

[2]We do not use the corpus-based retrieval score normalization as in the original implementations of *WIG* [176] and *NQC* [138]. Rather, we sum-normalize the entity retrieval score with respect to the scores of all entities in the result list following previous recommendations [138].

AutoCorrelation (AC) The auto-correlation predictor [48] (AC), which was proposed for document retrieval, measures the extent to which similar documents in the result list are assigned with similar retrieval scores. We use AC for the entity retrieval task as follows. First, the retrieval scores of the entities in the result list are normalized to have a zero mean and unit variance. Then, all entities in the list are assigned with a second score. This new ("regularized") score is the weighted average of the original (normalized) scores of the entity's k nearest neighbors in the list; k is a free parameter. Nearest neighbors are determined using the inter-entity similarity measures (*doc* or *type*) which also serve for weighting. The prediction value is the Pearson correlation between the original (normalized) scores in the list and the new scores. The resultant predictors, which differ by the inter-entity similarity measure employed, are denoted $AC_{rep=score;Sim(=,doc)}$ and $AC_{rep=score;Sim(=,type)}$. These predictors are based on the premise that "similar" entities should be assigned with similar retrieval scores. This prediction principle is a manifestation of the cluster hypothesis which was explored in Section 3.1.

Max Cluster Score (MCS) The AC predictor can assign high prediction values to result lists with very low (yet similar) retrieval scores. The *WIG* predictor assigns a high prediction value if the entities' scores at the top ranks of the list are high. However, *WIG* does not account for the extent to which similar entities are assigned with similar scores. Hence, to conceptually leverage the strengths of the two approaches, we present a novel prediction method (MCS).

The predictor uses nearest-neighbor clustering of the entity result list. Each entity and its k nearest neighbors in the list form a cluster. The score of a cluster is the geometric mean of the normalized retrieval scores of its constituent entities [128].[3] The maximal cluster score is the prediction value. The resultant predictors, $MCS_{rep=score;Sim(=,doc)}$ and $MCS_{rep=score;Sim(=,type)}$, use the *doc* and *type* inter-entity similarity measures, respectively, to create clusters. The prediction principle is that a result list which contains entities that are (i) similar to each other, and (ii) assigned with high retrieval scores, is likely to be effective.

3.2.3 Query performance prediction evaluation

3.2.3.1 Experimental setup

We performed experiments with the datasets of the INEX *entity ranking track* of 2007 and 2008 [44, 47]. These tracks used the English Wikipedia dataset from 2006. The tracks provide a total of 109 topics for the entity ranking task, which were originally used for training and testing. We use all of these queries in our

[3]Normalized retrieval scores are attained by a sum-normalization of the exponents of the original scores.

experiments.[4] The data is pre-processed using Lucene[5], including tokenization, stopword removal, and Porter stemming.

To measure prediction quality, we follow common practice in work on QPP for document retrieval [28]. We use the Pearson correlation between the prediction values assigned to a set of queries by a predictor and the ground-truth average precision (AP@1000) which is determined based on relevance judgements.[6]

To set the values of free parameters of predictors, we applied 100 tests of 2-fold cross validation performed over all queries. The resultant average prediction quality is reported. Statistically significant differences of prediction quality are determined using the two-tailed paired t-test computed over the folds using a 95% confidence level. Prediction quality (measured using Pearson correlation) serves as the optimization criterion in the learning phase. The 2-fold procedure enables to have enough queries (~55) in both the train and test sets so as to compute Pearson correlation in a robust manner. The free-parameter values of each predictor's version (*doc*, *type* and $doc \wedge type$) were learned separately.

Clarity and *QF* use the RM1 relevance model [93] which is constructed from maximum likelihood estimates of the entities' representations (*doc* or *type*). The exponent of the entities' retrieval scores (described below) serve for entity weighting. The number of terms used by RM1, and the number of top-retrieved entities used to construct it, are set to values in $\{10, 50, 100\}$ and $\{25, 50, 100\}$, respectively. QF's l_{qf} parameter is selected from $\{5, 10, 20, 30, 40, 50\}$.

The number of most highly ranked entities considered by *WIG* and *NQC*, n, is selected from $\{5, 10, 20, 30, 40, 50, 100\}$ and $\{10, 20, 30, 40, 50, 100, 500\}$, respectively. For *Cohesion*, *AC* and *MCS*, n is set to values in $\{10, 50, 100\}$. The number of nearest neighbors, k, used in the *AC* and *MCS* predictors, is selected from $\{4, 9\}$.

We predict the effectiveness of two lists, each contains 1000 entities, that are retrieved using effective methods [129]. The first, L_{Doc}, is created by applying a standard language-model-based approach upon the *doc* representation of entities. The score of entity e, represented by document e_x, with respect to query q is based on the cross entropy measure: $S_D(e) \stackrel{def}{=} -CE(p_q^{[0]}(\cdot)||p_{e_x}^{[100]}(\cdot))$. (Refer to the description of the inter-entity similarity measures in Section 3.2.1 for details regarding the language model notation used.) The second list, $L_{Doc;Type}$, is created by re-ranking L_{Doc} using a linear interpolation of two entity retrieval scores. The first is that used to create L_{Doc} (i.e., $S_D(e)$). The second is an entity-type-based score, $S_T(e)$. Specifically, it is the minus of the minimum (normalized) distance,

[4]We did not use the 2009 dataset since there are too few queries for learning free-parameter values of predictors.

[5]http://lucene.apache.org/core/

[6]The performance for queries of the 2008 track was originally evaluated using extended inferred average precision (xinfAP) [163]. We found that the standard AP measure is 99.99% correlated with xinfAP for the retrieval methods we use. Hence, for consistency with the queries used in 2007, AP was used in all experiments.

over Wikipedia's category graph, between a category in the query target type and a category among those associated with e. The interpolated score assigned to e is: $\lambda \log \frac{\exp(S_D(e))}{\sum_{e' \in L} \exp(S_D(e'))} + (1 - \lambda) \log \frac{\exp(S_T(e)]}{\sum_{e' \in L} \exp(S_T(e'))}$; λ is a free parameter set to 0.5. The rest of the technical details regarding the implementation of the retrieval methods follow those in [129].

3.2.3.2 Experimental results

Table 3.4 presents the prediction quality numbers. Our first observation is that the most effective pre-retrieval predictors are outperformed by the most effective post-retrieval predictors, as reported for document retrieval [28]. Also, the *Clarity* predictors are less effective than most other post-retrieval predictors. QF, which is a state-of-the-art predictor for document retrieval, is outperformed (often substantially) by quite a few other post-retrieval predictors. WIG and NQC, which analyze retrieval scores, are highly effective, similarly to the case for document retrieval [28].

The *Cohesion* predictor posts poor prediction quality when using the *doc* inter-entity similarity measure. This finding is in accordance with those reported for document retrieval [28]. However, the prediction quality is relatively high when using the *type* inter-entity similarity measure. Thus, an entity result list which is cohesive in terms of the categories of the entities it contains is somewhat likely to be effective. In contrast to the case for *Cohesion*, for the AC predictor the *doc* inter-entity similarity measure is more effective than the *type* measure. This finding could potentially be attributed to the sparseness of the *type* measure. That is, in some cases an entity might not share categories with other entities in the list and hence the inter-entity similarity is 0. We use entity IDs to break similarity ties.

Predictors employed with both the $rep = doc$ and $rep = type$ representations are in most cases more effective when using the former than the latter. Yet, in quite a few cases (e.g., for *maxIDF* and *Clarity*), using both representations ($rep = doc \wedge type$) is superior to using either.

The prediction quality for almost all predictors is higher for the L_{Doc} list than it is for the $L_{Doc;Type}$ list. Recall that $L_{Doc;Type}$ is a re-ranked version of L_{Doc} created by interpolation of two entity scores. The first is based on the entity's document and the second is based on the entity's categories. However, the category-based information (distance in the Wikipedia category graph) is different than that used by the prediction methods (terms in categories' names), and therefore the prediction quality for $L_{Doc;Type}$ might be lower. We hasten to point out, however, that some of the prediction quality numbers for $L_{Doc;Type}$ are quite high and competitive with those for L_{Doc}; e.g., for $WIG_{rep=score}$ and $NQC_{rep=score}$ that use retrieval scores.

Our novel MCS predictor is the most effective for the L_{Doc} list when using the *doc* inter-entity similarity measure ($MCS_{rep=score;Sim(=,doc)}$); this predictor outperforms to a statistically significant degree all other predictors. Furthermore, $MCS_{rep=score;Sim(=,type)}$ outperforms to a statistically significant degree all predic-

Predictor	$\mathbf{L_{Doc}}$	$\mathbf{L_{Doc;Type}}$
$avgIDF_{rep=doc}$	$0.555^{d,t}_{\wedge}$	$0.441^{d,t}_{\wedge}$
$avgIDF_{rep=type}$	$0.297^{d,t}_{\wedge}$	$0.248^{d,t}_{\wedge}$
$avgIDF_{rep=doc\wedge type}$	$0.523^{d,t}_{\wedge}$	$0.414^{d,t}_{\wedge}$
$sumIDF_{rep=doc}$	$0.070^{d,t}_{\wedge}$	$-0.002^{d,t}_{\wedge}$
$sumIDF_{rep=type}$	$0.042^{d,t}_{\wedge}$	$0.106^{d,t}_{\wedge}$
$sumIDF_{rep=doc\wedge type}$	$0.100^{d,t}_{\wedge}$	$0.105^{d,t}_{\wedge}$
$maxIDF_{rep=doc}$	$0.475^{d,t}_{\wedge}$	$0.280^{d,t}_{\wedge}$
$maxIDF_{rep=type}$	$0.254^{d,t}_{\wedge}$	$0.191^{d,t}_{\wedge}$
$maxIDF_{rep=doc\wedge type}$	$0.489^{d,t}_{\wedge}$	$0.301^{d,t}_{\wedge}$
$avgVarTF.IDF_{rep=doc}$	$0.547^{d,t}_{\wedge}$	$0.444^{d,t}_{\wedge}$
$sumVarTf.IDF_{rep=doc}$	$0.395^{d,t}_{\wedge}$	$0.294^{d,t}_{\wedge}$
$maxVarTf.IDF_{rep=doc}$	$0.532^{d,t}_{\wedge}$	$0.377^{d,t}_{\wedge}$
$Clarity_{rep=doc}$	$0.370^{d,t}_{\wedge}$	$0.295^{d,t}_{\wedge}$
$Clarity_{rep=type}$	$0.303^{d,t}_{\wedge}$	$0.279^{d,t}_{\wedge}$
$Clarity_{rep=doc\wedge type}$	$0.369^{d,t}_{\wedge}$	$0.312^{d,t}_{\wedge}$
$WIG_{rep=score}$	$0.651^{d}_{\wedge}$	$\mathbf{0.623}^{d,t}_{\wedge}$
$NQC_{rep=score}$	$0.600^{d,t}_{\wedge}$	$0.578^{d,t}_{\wedge}$
$QF_{rep=doc}$	$0.437^{d,t}_{\wedge}$	$0.410^{d,t}_{\wedge}$
$Cohesion_{Sim(=,doc)}$	$-0.026^{d,t}_{\wedge}$	$-0.106^{d,t}_{\wedge}$
$Cohesion_{Sim(=,type)}$	$0.508^{d,t}_{\wedge}$	$0.403^{d,t}_{\wedge}$
$Cohesion_{Sim(=,doc\wedge type)}$	$0.360^{d,t}_{\wedge}$	$0.257^{d,t}_{\wedge}$
$AC_{rep=score;Sim(=,doc)}$	$0.475^{d,t}_{\wedge}$	$0.378^{d,t}_{\wedge}$
$AC_{rep=score;Sim(=,type)}$	$0.418^{d,t}_{\wedge}$	$0.319^{d,t}_{\wedge}$
$AC_{rep=score;Sim(=,doc\wedge type)}$	$0.468^{d,t}_{\wedge}$	$0.360^{d,t}_{\wedge}$
$MCS_{rep=score;Sim(=,doc)}$	**0.665**	0.563
$MCS_{rep=score;Sim(=,type)}$	0.650	0.596
$MCS_{rep=score;Sim(=,doc\wedge type)}$	0.591	0.502

Table 3.4: Prediction quality. The best result in a column per a result list is boldfaced. 'd', 't' and '$\wedge$' mark statistically significant differences with $MCS_{rep=score;Sim(=,doc)}$, $MCS_{rep=score;Sim(=,type)}$ and $MCS_{rep=score;Sim(=,doc\wedge type)}$, respectively.

tors except for WIG. For the $L_{Doc;Type}$ list, $MCS_{rep=score;Sim(=,type)}$ and $MCS_{rep=score;Sim(=,doc)}$ are the second and fourth best, respectively. While the former outperforms all predictors, except for WIG, to a statistically significant degree, it is outperformed by WIG in a statistically significant manner. All in all, these findings attest to the merits of our MCS predictor that relies on the cluster hypothesis.

Chapter 4
Related Work - Utilizing Entities for Ad Hoc Document Retrieval

We now turn to address a very fundamental task in information retrieval: *ad hoc document retrieval*. The goal is estimating the relevance of *documents* in a corpus with respect to a user's information need, formulated using a query. In Chapter 5 we suggest two novel types of surface level entity-based representations for addressing the task of ad hoc document retrieval. By "surface level" we refer to representations that are based only on entities marked in the text and on terms appearing in it. In Chapter 6, we suggest novel methods utilizing inter-entity similarity estimates for inducing entity-based query models. Such query models can be viewed as expanded query forms. Both these works address two fundamental challenges regarding the use of entities for ad hoc document retrieval: (1) how can surface level entity-based query and documents representations be effectively utilized for document retrieval? (2) how can entity associated information be utilized for inducing entity-based query models?

In this chapter we survey past work related to these two challenges. In Section 4.1, we present approaches for marking entities in a text to create surface level entity-based representations. Then, in Section 4.2 we present document retrieval methods that make use of surface level entity-based representations. We also describe methods utilizing surface level entity-based representations for additional tasks such as clustering and classification. In Section 4.3 we describe past work on methods utilizing entities for inducing query models. Methods utilizing inter-*term* similarities for inducing query models are also surveyed. Finally, additional approaches for utilizing entities for document retrieval are presented in Section 4.4.

4.1 Creating Entity-Based Representations

Entities can be associated with texts manually [135, 140] or automatically [4, 51, 69, 147]. The associated entities can either have mentions in the text [4, 76] or can somehow be related to the text [51]. The methods we propose in Chapters 5 and 6 utilize information about term sequences in a text that are marked as entities by *some* entity-linking tool [57, 115]. In the following, we describe the entity linking task as well as common approaches for addressing it.

Entity linking systems are aimed at identifying, in an input text, short and meaningful term sequences, and marking them with unambiguous identifiers which correspond to entities in an entity repository [38]. An entity linking tool is assumed to provide a confidence level for each entity markup.

Many methods have been proposed for addressing the entity linking task (e.g., [41, 57, 56, 71, 84, 106, 110, 113]). In this work, we focus on methods utilizing Wikipedia as the entity repository [57, 113]. Wikipedia is considered a high quality

entity repository due to its wide coverage and content, composed of both structured and unstructured information.

There are two main stages in the entity linking process. The goal of the first stage is identifying term sequences that are candidates for linking, and creating a list of candidate entities that can be associated with each term sequence. The goal of the second stage is disambiguating the meaning of a candidate term sequence, i.e., selecting the linked entity.

The first stage is usually addressed by using anchor texts appearing in Wikipedia as candidate term sequences and using the respective Wikipedia page as candidate entity for the disambiguation process [57]. Alternatively, named entity recognition tools (e.g., [58]), designed for identifying mentions of named entities (e.g., people, locations) in text, are used [41].

The second stage, entity disambiguation, is addressed by utilizing "global" and "local" features which are estimates of the probability that a term sequence should be marked with a given candidate entity [57, 84]. By "global features" we refer to features utilizing the training corpus information which is associated with a candidate entity. For example, the number of times a term sequence is marked by a specific entity ID out of all its markups in the training set is often used as a global feature. By "local features" we refer to features utilizing information regarding the specific context of a marked term sequence. For example, the semantic relatedness between a candidate entity and additional candidate entities appearing in the text sequence context is used as a local feature [57, 84].

In our work we use the TagMe entity-linking tool[1] which was shown to be highly effective and efficient in comparison to other publicly available entity-linking systems [38]. We also use the Wikifier entity-linking tool[2][30, 38] to evaluate whether our proposed methods are robust with respect to the entity linking tool utilized for creating the entity markups.

4.2 Using Entity-Based Representations for Document Retrieval and Additional Related Tasks

There are several works on devising surface level entity-based document and query representations for document retrieval [4, 51, 69, 83, 140, 147, 162]. The findings about the merits of these representations have been inconclusive. The few cases where the representations were shown to be somewhat effective for retrieval were when entity markups were devised in extreme care and were of very high quality [4, 51, 162]. Also, many of these works were focused on vector space models.

There is a recent work on utilizing bag-of-entities representations, induced by marking queries and documents using entity linking tools, for document retrieval [157]. Methods utilizing the appearance of query entities and their counts in doc-

[1] `tagme.di.unipi.it`

[2] `cogcomp.cs.illinois.edu/page/demo_view/Wikifier`

uments were proposed. These methods were shown to be effective with respect to a standard term-based unigram language model retrieval [92].

In contrast to this past work, the models we suggest in Chapter 5 are language models that serve for retrieval in the language modeling framework. In addition, in contrast to all previously proposed representations [4, 51, 69, 83, 140, 147, 157, 162], our language models account, simultaneously, for the uncertainty in the entity-markup process, and the balance between using term-based and entity-based information. Consequently, a highly important aspect that further differentiates our approach from related work is the effective utilization of high recall, noisy, entity markups. Finally, we demonstrate the clear merits of using our models for retrieval. For instance, we show that the performance of our proposed models significantly transcends that of the sequential dependence model (SDM) [77, 111]. Integrating the language models with SDM yields further performance improvements.

In some studies, concepts (entities) in verbose queries were automatically weighted [2, 19, 20, 85]. In contrast to our approach, weights (confidence levels) of entities in documents were not accounted for. We demonstrate the importance of accounting for the confidence level of entity markups in both queries and documents.

Entity-based vector space document representations have been used for clustering [17, 72, 73, 74, 75, 76] and classification [60, 148]. Differential weights have been assigned to different feature types (e.g., terms vs. entities) either at the representation level [17] or at the inter-document similarity score level [73, 74]. In a conceptually similar vein, the methods we devise for *document retrieval* assign different importance weights to terms and entities at the language model level or at the retrieval score level. In some of the representations proposed for clustering [72, 74], entities with low strength of relationship with other entities in the text were pruned so as to improve representation quality. One of our suggested language models, which does not consider entity markups with a confidence level lower than some threshold, is inspired by this approach. Yet, another language model that we devise, and which accounts for all entity markups and weighs their contribution by their confidence levels, is shown to yield better retrieval performance. Furthermore, in contrast to the work on clustering just mentioned, we show that using entity markups with low confidence level (i.e., utilizing high recall entity markup) is actually very important for attaining effective retrieval provided that term-based information is also utilized to a sufficient extent.

Motivated by our empirical findings regarding the effectiveness of using entity-based representations for retrieval, and by work on the use of entity-based representations for clustering, we demonstrate the merits of using our language models for cluster-based document retrieval. Using entity-based representations for this task is novel to this study.

There are language models that integrate word phrases and named entities based on their association with predefined classes [86, 94]. In contrast to our language models, which are not based on such classes, these language models were not designed and used for document retrieval.

4.3 Entity-Based Query Models

There is much work on inducing query models using entity-based information [7, 23, 42, 99, 108, 109, 156, 160]. The most common approach relies on identifying entities that explicitly appear in the query [23, 42, 160] or that are related to it [7, 42, 99, 109, 156]. Terms associated with these entities are used for inducing term-based query models. Specifically, different features modeling associations between these terms and the entities related to the query are used for estimating the query model probabilities.

The query model probabilities are essentially estimates of the term relevance, i.e., the probability that a term is relevant with respect to the information need. Some methods for *learning term relevance*, which is then used for inducing a term-based query model, were proposed [23, 156, 158]. We elaborate on the concept of "term relevance" below (see Section 4.3.1).

In contrast to most of these query-model induction methods, we use *both* terms and entities (i.e., tokens marked by some entity linking tool and are identified by an entity ID) for inducing query models. Specifically, entity-related information is *not* used for estimating the probabilities assigned to terms in our proposed models. Entities serve as tokens that are assigned query model probabilities.

Dalton et al. [42] used entity-based representations for inducing query models, i.e., entity ID's marked in texts were assigned query model probabilities. In their work, multiple term-only and entity-only query models, induced by utilizing various types of information (in addition to entity IDs), were fused using a learning to rank method. In contrast to their work, we explore the merits of utilizing a specific type of entity associated information for inducing entity-based query models, that is, inter-entity similarity estimates. We experiment with various inter-entity similarity measures and query model induction methods to gain a deep understanding regrading the merits of using inter-entity similarities for query models induction.

There is much work on utilizing similarities between *terms* (in contrast to entities) for inducing term-based query models [35, 49, 63, 80, 91, 134, 167, 166, 168]. Earlier methods utilized co-occurrence statistics such as mutual information to determine which terms are strongly associated with the query terms [35, 63, 80]. Additional semantic relations such as synonymy and general word association were also utilized [35].

The recent success of learning semantically meaningful term embeddings by applying Word2Vec [114] or GloVe [121] has led to the development of methods utilizing term embeddings for inducing term-based query models [49, 166, 168]. The basic approach is assigning high model probabilities to terms most similar to the query [91, 134, 166]. The induced query models were shown to be effective for document retrieval, i.e., using them improved retrieval effectiveness in comparison to using the queries alone. Still, using the state-of-the-art query relevance model results in better retrieval performance [93]. It has been shown that integrating the relevance model with a query model induced by utilizing term embeddings yields retrieval performance that transcends that of using each of the models alone

[91, 166]. Moreover, adapting the word embeddings learning process to the ad hoc retrieval task [49, 168], by locally training word embeddings [49] or by changing the learning goal [168], has been shown to result in significant improvement in retrieval effectiveness.

Some of the works on utilizing inter-term similarities for inducing term-based query models are very similar in spirit to ours [8, 103, 109, 166]. In contrast to these works we use inter-*entity* similarity estimates to induce *entity*-based query models. In addition, we experiment with a variety of similarity measures as described below.

The inter-entity similarity measures we use utilize different types of entity associated information (see details in Section 6.3). First, query dependent [99, 142] and independent [103, 152] textual similarity between the pages of two compared entities is utilized. Also, we use shared incoming and outgoing links for the two Wikipedia pages that represent entities, so as to estimate their similarity [153]. We use co-occurrence information by considering the mutual information of an entity pair. Finally, the similarity between the embedding-based representations is utilized. We train a continuous bag-of-word (CBOW) model of Word2Vec[3] for learning entity embeddings, using various collections.

Textual similarity between entity pages as well as co-occurrence information were utilized for estimating inter-entity similarities that are used for document retrieval [103, 99, 155, 158]. Entity embeddings were used for comparing entities in a fuzzy match retrieval method proposed in past work [159]. These embeddings were trained based on the entities' neighbors in the Freebase[4] knowledge graph. In our work, we utilize Wikipedia, which is a semi-structure knowledge base and therefore the text of the repository instead of its graph structure is used for learning entity embeddings. We are not familiar with works utilizing entities' shared links or query dependent textual similarities for inducing query models.

4.3.1 Estimating token (entity or term) relevance

In Chapter 6 we propose the "second-order cluster hypothesis", which is inspired by the cluster hypothesis for document retrieval [144]. This hypothesis is novel to this study. The hypothesis is: *closely associated entities tend to be relevant to the same requests*. It is assumed that the type of retrieved item (document) can be different from the type of the item for which the hypothesis is stated (entities).

Evaluating the second-order cluster hypothesis requires relevance estimates for entities. Some previously proposed methods for estimating term relevance were proposed [23, 25, 156]. Inspired by these methods we propose to estimate *entity* relevance by directly evaluating the effectiveness of using it for inducing a query model. In contrast to past work we only estimate the relevance of *entities* to an information need. We use automatically generated relevance judgments for entities to test the second-order cluster hypothesis.

[3]https://code.google.com/archive/p/word2vec/

[4]https://developers.google.com/freebase/

4.4 Additional Methods Utilizing Entities for Document Retrieval

There are works on utilizing explicit term-based and entity-based representations for fuzzy query-document match estimation [53, 158, 159]. A different line of work is on projecting queries and documents onto latent entity space and comparing them on that space [51, 103, 155]. In these works, queries and documents are represented by external terms and entities which they do not contain. Also, auxiliary information about entities from the entity repository is used.

In contrast to these approaches, our work in Chapter 5 is focused on exact match between entity-based query and document representations. Our proposed representations utilize the entity markups simply as tokens with confidence levels, and do not use auxiliary information. Our work in Chapter 6 is focused on exact match between entity-based query models, induced by utilizing inter-entity similarities, and document models. In this work we do use auxiliary information that is associated with entities. However, this information is utilized by a conceptually different retrieval framework. In addition, we use this information for a specific purpose which is inducing inter-entity similarities. Specifically, we explore the retrieval merits of using inter-entity similarity estimates for inducing entity-based query models.

Chapter 5
Document Retrieval Using Entity-Based Language Models

In this chapter we address a fundamental challenge regarding the use of entity-based information for document retrieval. We study whether using *surface level* entity-based query and document representations can help to improve retrieval effectiveness. By "surface level" we refer to representations based only on terms in the text and *markups* of entities in it, along with raw corpus-based occurrence statistics. This is in contrast to expansion-based and projection-based representations that utilize also terms and entities related to those (marked) in the text and which often use auxiliary information about entities from the entity repository; e.g., textual descriptions of entities, entities' categories and inter-entity relations [160, 109, 120, 23, 42, 99, 156, 95, 103, 155]. Put in simpler words, the question we address is *whether the markup of entities in a query and documents is, by itself, sufficient information for improving retrieval effectiveness*.

The reason for addressing the question just posed is two fold. First, it will shed light on the effectiveness of using entities in their most basic capacity; that is, special tokens marked in queries and documents. Indeed, findings in past work on ad hoc retrieval regarding the merits of using surface level entity-based representations are inconclusive [69, 147, 162, 4, 51]. Second, such representations can be naturally used in existing retrieval approaches and tasks to improve performance; e.g., query expansion and cluster-based document retrieval as we show in this chapter.

There are various potential merits in using surface level entity-based representations. For example, these can help to cope with the vocabulary mismatch problem; e.g., the entity *United States of America* can have different expressions in the text, including, "U.S.", "USA", "United States" and more. Furthermore, expressions of entities in the text are variable-length n-grams that bear semantic meaning. Thus, entities can be used for effective modeling of term proximity information which goes beyond using fixed-length n-grams.

An important challenge in inducing entity-based representations is accounting for the uncertainty inherent in the entity-markup process (a.k.a. entity linking); that is, associating term sequences with entities in a repository. Specifically, a term sequence can potentially be associated with multiple entities; e.g., the term "Lincoln" can be associated with the U.S. president, the car, the 2012 movie, etc. The uncertainty in entity linking has significant impact on retrieval effectiveness as we show in this chapter.

We present novel types of entity-based language models which consider *both* single terms in the text and term sequences marked as entities by an existing entity-linking tool. These language models are induced from the query and documents in the corpus and serve for retrieval in the language modeling framework. The main novelty of these language models is accounting, 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. We demonstrate the importance of accounting for the mutual effects of these two aspects. For example, we show that using high recall entity markup, which is quite noisy, can help to significantly improve retrieval effectiveness if the noise is "balanced" by sufficient utilization of term-based information.

Empirical evaluation demonstrates the merits of using our entity-based language models for retrieval. The performance significantly transcends that of a state-of-the-art term proximity method: the sequential dependence model (SDM) [111, 77]. Integrating the language models with SDM yields further performance improvements. The language models are also effective for two additional retrieval paradigms: cluster-based document retrieval and query expansion.

5.1 Retrieval Framework

In what follows we present ad hoc document retrieval methods that rank documents in a corpus D in response to query q. The methods utilize information about entities mentioned in the query and in documents.

To mark entities in texts, we use *some* entity-linking tool that utilizes a repository (e.g., Wikipedia or Freebase) where entities have unique IDs. The entity-linking tool takes as input a text, query or document in our case, and marks variable length sequences of terms as *potential* entities in the repository. The entity markup of a term sequence is composed of entity ID and a confidence level in $[0, 1]$. The confidence level reflects the likelihood that the term sequence corresponds to the entity. The confidence level relies on the term sequence and its context; e.g., its neighboring terms or other term sequences marked as entities [57, 115]. Using high confidence level results in high precision entity markup while low confidence level results in high recall.

We assume that each position in a given text can be part of at most a single term sequence that is marked as an entity; i.e., the entity markups do not overlap. A specific occurrence of a term sequence in a text cannot be marked with more than one entity. Yet, a term sequence can appear several times in a text with different entity markups as the markups depend on the context of the sequence. Details of the entity linking tools we use are provided in Section 5.2.1.

The retrieval methods we present in Section 5.1.2 use entity-based query and document language models. We now turn to define these language models.

5.1.1 Entity-based language models

We define unigram entity-based language models over a token space $\mathcal{T}$; i.e., tokens are generated by the language model independently of each other. The token space,

$$\mathcal{T} \stackrel{def}{=} \mathcal{V} \cup \mathcal{E} \tag{5.1}$$

is composed of the set $\mathcal{V}$ of all terms in the corpus D and the set $\mathcal{E}$ of entities in the entity repository which were marked at least once in a document in D with *any* confidence level.

The language models we devise rely on a definition of *pseudo counts* for tokens. Two definitions of pseudo counts will be presented in Sections 5.1.1.1 and 5.1.1.2. Let $pc(t, x)$ be the pseudo count of token t ($\in \mathcal{T}$) in the text or text collection x. We define the *pseudo length* of x as:

$$pl(x) \stackrel{def}{=} \sum_{t \in \mathcal{T}: pc(t,x) > 0} pc(t, x).$$

The maximum likelihood estimate (MLE) of token t ($\in \mathcal{T}$) with respect to x is:

$$\theta_x^{MLE}(t) \stackrel{def}{=} \frac{pc(t, x)}{pl(x)}. \tag{5.2}$$

The MLE can be smoothed using Dirichlet priors [172]:

$$\theta_x^{Dir}(t) \stackrel{def}{=} \frac{pc(t, x) + \mu \theta_D^{MLE}(t)}{pl(x) + \mu}; \tag{5.3}$$

μ is a smoothing parameter.

We next describe two types of language models defined over $\mathcal{T}$ and induced using Equations 5.2 and 5.3. The language models differ by the definition of pseudo counts for tokens.

5.1.1.1 Hard confidence-level thresholding

The hard confidence-level thresholding language model, **HTLM** in short, is based on *fixing* a threshold τ ($\in [0, 1]$) for entity markups. Entity-based information is used only for entity markups with confidence level $\geq \tau$. In contrast, *every* term occurrence in a text, including those in entity markups with a confidence level $< \tau$, is accounted for.

To formally define a HTLM using Equations 5.2 and 5.3, we have to define pseudo counts for tokens from $\mathcal{T}$ in a text or text collection x. To that end, we lay down a few definitions. If t ($\in \mathcal{T}$) is a term, then $c_{term}(t, x)$ is the number of occurrences of t in x. Let $\mathcal{M}(x)$ denote the set of all entity markups in x; i.e., all occurrences of term sequences in x that were marked as entities with some confidence level. For a markup m ($\in \mathcal{M}(x)$), $E(m)$ is the entity and $\rho(m)$ is the confidence level. The equivalence relation $t_1 \equiv t_2$ holds iff the entity tokens t_1 and t_2 are identical (i.e., have the same ID). The pseudo count of t ($\in \mathcal{T}$) in x is based on (i) the raw count of t in x if t is a term; and, (ii) the number of entity markups of t in x with a confidence level $\geq \tau$ if t is an entity. Formally,

$$pc_{HTLM;\tau}(t, x) \stackrel{def}{=} \tag{5.4}$$
$$\begin{cases} \lambda c_{term}(t, x) & \text{if } t \in \mathcal{V}; \\ (1 - \lambda) \sum_{m \in \mathcal{M}(x): E(m) \equiv t} \delta[\rho(m) \geq \tau] & \text{if } t \in \mathcal{E}; \end{cases}$$

λ ($\in [0, 1]$) is a free parameter which controls the relative importance attributed to term and entity tokens; δ is Kronecker's delta function: for statement s, $\delta[s] = 1$ if s is true and $\delta[s] = 0$ otherwise.

We note that using a Dirichlet smoothed HTLM (i.e., using Equation 5.4 in Equation 5.3) can still result in assigning zero probability to some tokens in $\mathcal{T}$. These are entities with no corresponding markup of a term sequence in the corpus with confidence level $\geq \tau$. We re-visit this point below.

If we set $\lambda = 1$ in Equation 5.4, then the resultant HTLM reduces to a standard unigram term-based language model. Setting $\lambda = 0$ results in **HTEntLM** which is a unigram language model that assigns non-zero probability *only* to entities: if the MLE from Equation 5.2 is used, then these are the entities with at least one markup in x with a confidence level $\geq \tau$; if the Dirichlet smoothed language model is used (Equation 5.3), then these are the entities with at least one markup in the corpus with a confidence level $\geq \tau$.

5.1.1.2 Soft confidence-level thresholding

A potential drawback of HTLM is committing to a specific threshold τ for entity markups. That is, information about entity markups with confidence level lower than τ is ignored. Furthermore, all entity markups with confidence level $\geq \tau$ are counted equally as their confidence levels are ignored.

Thus, we now turn to present a soft confidence-level thresholding language model, **STLM**. STLM accounts for all markups of an entity and weighs them by the corresponding confidence levels. Specifically, the pseudo count of t ($\in \mathcal{T}$) in the text or text collection x is defined as:

$$pc_{STLM}(t, x) \stackrel{def}{=} \begin{cases} \lambda c_{term}(t, x) & \text{if } t \in \mathcal{V}; \\ (1 - \lambda) \sum_{m \in \mathcal{M}(x): E(m) \equiv t} \rho(m) & \text{if } t \in \mathcal{E}; \end{cases} \tag{5.5}$$

λ ($\in [0, 1]$) is a free parameter that, as in HTLM, controls the relative importance attributed to term and entity tokens. Thus, STLM addresses the uncertainty inherent in the entity linking process by using *expected* entity occurrence counts; the corresponding confidence levels serve for occurrence probabilities. These expected counts are then integrated with deterministic term counts.

If we set $\lambda = 1$ in Equation 5.5, then STLM reduces to a standard unigram term-based language model as was the case for HTLM. Setting $\lambda = 0$ results in **STEntLM**. This language model assigns a non-zero probability only to entities that have at least one markup (with any confidence level) in x when using the MLE (Equation 5.2) or in the corpus when using the Dirichlet smoothed language model (Equation 5.3). We note that in contrast to the case for HTLM, there is no token in $\mathcal{T}$ that is assigned a zero probability by a Dirichlet smoothed STLM.

5.1.2 Retrieval models

We rank document d by the cross entropy between the language models induced from the query (q) and d [92]:

$$CE\left(\theta_q \mid\mid \theta_d\right) = -\sum_{t \in \mathcal{T}} \theta_q(t) \log \theta_d(t); \tag{5.6}$$

higher values correspond to decreased similarity.

Equation 5.6 is instantiated using the entity-based language models described in Section 5.1.1. Following common practice [170], we use an unsmoothed maximum likelihood estimate for the query language model (Equation 5.2) and a Dirichlet smoothed document language model (Equation 5.3). We obtain four retrieval methods : **HT**[1], **HTOEnt**, **ST** and **STOEnt**[2], which utilize the HTLM, HTEntLM, STLM and STEntLM language models, respectively. HT and ST utilize entity and term tokens, while HTOEnt and STOEnt utilize only entity tokens, hence the "O" in the methods names.

5.1.2.1 Score-based fusion

The HTLM and STLM language models integrate term-based and entity-based information at the *language model level*. Hence, the query-document comparison in Equation 5.6 simultaneously accounts for the appearance of the query terms and entities in a document.

An alternative approach is integrating term and entity information at the *retrieval score level*. Inspired by approaches in the vector-space model [147], and in work on using a latent entity space [103], we consider a method that fuses document retrieval scores produced by utilizing, *independently*, term-only (θ_x^{term}) and entity-only (θ_x^{ent}) language models induced from text x. Document d is scored by:

$$\lambda CE\left(\theta_q^{term} \mid\mid \theta_d^{term}\right) + (1-\lambda) CE\left(\theta_q^{ent} \mid\mid \theta_d^{ent}\right); \tag{5.7}$$

[1]In HT, the *same* confidence-level threshold, τ_d, is used for all documents; the query threshold, τ_q, can be different from τ_d. Hence, an entity token assigned a non-zero probability by θ_q could be assigned a zero probability by θ_d; e.g., an entity marked in q with a confidence level $\geq \tau_q$ but with no markup in the corpus with confidence level $\geq \tau_d$. In these cases, we zero the probability assigned to the entity token by θ_q to avoid a $\log 0$ in Equation 5.6. This is common practice in addressing term tokens that appear in a query but not in any document in the corpus.

[2]HTOEnt and STOEnt rely only on entity tokens. If all entities in $\mathcal{E}$ are assigned a zero probability by the unsmoothed query language model, then no documents are retrieved. This can happen for example when inducing HTEntLM from the query with a high confidence-level threshold or inducing a STEntLM from a query which has no entity markups.

Table 5.1: TREC data used for experiments.

corpus	# of docs	data	queries
AP	242,918	Disks 1-3	51 − 150
ROBUST	528,155	Disks 4-5 (-CR)	301 − 450, 601 − 700
WT10G	1,692,096	WT10g	451 − 550
GOV2	25,205,179	GOV2	701 − 850
ClueB ClueBF	50,220,423	ClueWeb09 (Cat. B)	1 − 200

the λ parameter balances the score fusion[3]. The query language models are unsmoothed maximum likelihood estimates (Equation 5.2) and the document language models are Dirichlet smoothed (Equation 5.3).

Instantiating Equation 5.7 with an entity-only language model, HTEntLM or STEntLM, and with a standard unigram term-based language model yields the **F-HT** and **F-ST** methods, respectively. These are conceptually highly similar to the HT and ST methods which integrate term-based and entity-based information at the language-model level. However, HT and ST use a single smoothing parameter for both term and entity tokens (see Equation 5.3) while F-HT and F-ST can use a different smoothing parameter for each as they utilize separately term-only and entity-only language models. We could have used different smoothing parameters for entity and term tokens under the same language model, e.g., by applying term-specific smoothing [70], but we leave this exploration for future work.

5.2 Evaluation

5.2.1 Experimental setup

Experiments were conducted using the TREC datasets specified in Table 6.1. AP and ROBUST are mostly composed of news articles. WT10G is a small, noisy, Web collection. GOV2 is a much larger Web collection composed of high quality pages crawled from the .gov domain. ClueB is the English part of the Category B of the ClueWeb 2009 Web collection. ClueBF was created from ClueB by filtering from rankings suspected spam documents: those assigned a score below 50 by Waterloo's spam classifier [37].

Data processing Titles of TREC topics served for queries. Tokenization and Porter stemming were applied using the Lucene toolkit (lucene.apache.org) which was used for experiments. Stopwords on the INQUERY list were removed from queries but not from documents.

[3]The λ in the score-based fusion model has a conceptually similar role to that of λ in STLM and HTLM: balancing the use of term-based and entity-based information.

Unless otherwise specified, the **TagMe** entity-linking tool (`tagme.di.unipi.it`) is used to annotate queries and documents. TagMe uses Wikipedia (a July 2014 dump) as the entity repository, and was shown to be highly effective and efficient in comparison to other publicly available entity-linking systems [38]. In Section 5.2.2.1 we also show the effectiveness of our methods using the **Wikifier** entity-linking tool[4] [30, 38]. Wikifier was applied with an efficient configuration claimed to yield baseline entity linking effectiveness.

TagMe and Wikifier cannot process very long texts. Thus, we split documents into non-overlapping term-window passages. We terminate a passage at the first space that appears at least 500 characters after the beginning of the previous passage. We let the tools mark the passages independently. The tools are applied on the non-stemmed and non-stopped queries and documents. Entity markup of a term sequence includes an entity ID and a confidence level (in $[0, 1]$). We scan each text left to right and remove overlapping entity markups so that each position can be part of at most a single markup. If two markups overlap, we select the one with the higher confidence level. We break ties of confidence levels by selecting the markup which starts at the leftmost position.

Baselines We use standard term-based unigram language model retrieval [92], denoted **TermsLM**, for reference. This is a special case of the HT, ST, F-HT and F-ST methods with $\lambda = 1$. Documents are ranked by the cross entropy between the unsmoothed (MLE) query language model and Dirichlet smoothed document language models.

The **HTCon** method is a special case of HT with $\lambda = 0.5$ and $\tau_q = \tau_d = 0$ (τ_q and τ_d are the query and document thresholds, respectively). HTCon accounts *uniformly* for all entity mentions, and attributes the same importance to term and entity tokens. HTCon is conceptually reminiscent of methods representing documents and queries using concepts (e.g., from Wordnet) by concatenating with equal weights term-based and concept-based vector-space representations [140, 69, 147]. Accordingly, we consider **F-HTCon**: a special case of F-HT with $\lambda = 0.5$ and $\tau_q = \tau_d = 0$.

Additional baseline is the state-of-the-art sequential dependence model, **SDM**, from the Markov Random Field framework which utilizes term proximities [111, 77]. The comparison with SDM, and its integration with our STLM is presented in Section 5.2.2.3.

Evaluation measures and free-parameters Mean average precision at cutoff 1000 (MAP), precision of the top 10 documents (p@10) and NDCG@10 (NDCG) serve as evaluation measures. Statistically significant performance differences are determined using the two-tailed paired t-test with a 95% confidence level.

The free parameter values of *all* retrieval methods are set using 10-fold cross validation performed over the queries in a dataset. Query IDs are used to create

[4] `cogcomp.cs.illinois.edu/page/demo_view/Wikifier`

Table 5.2: Comparison of methods instantiated from Equation 5.6 using term-only (TermsLM) and entity-based language models. Bold: the best result in a row. 't', 'h', 'o', 'c' and 's' mark statistically significant differences with TermsLM, HT, HTOEnt, HTCon and ST, respectively.

		TermsLM	HT	HTOEnt	HTCon	ST	STOEnt
AP	MAP	20.9	23.1^{t}	$15.6^{t,h}$	22.5_{o}	$\mathbf{23.5^{t}_{o,c}}$	$17.5^{t,h}_{o,c,s}$
	p@10	39.1	$\mathbf{44.2^{t}}$	36.0^{h}	43.4^{t}_{o}	43.8^{t}_{o}	$38.3^{h}_{c,s}$
	NDCG	40.4	45.3^{t}	37.6^{h}	44.7^{t}_{o}	$\mathbf{45.5^{t}_{o}}$	$39.6^{h}_{c,s}$
ROBUST	MAP	25.0	$\mathbf{28.1^{t}}$	$19.1^{t,h}$	27.4^{t}_{o}	$\mathbf{28.1^{t}_{o,c}}$	$21.4^{t,h}_{o,c,s}$
	p@10	42.2	$\mathbf{45.5^{t}}$	$35.7^{t,h}$	45.0^{t}_{o}	45.3^{t}_{o}	$38.0^{t,h}_{o,c,s}$
	NDCG	43.5	$\mathbf{47.1^{t}}$	$36.9^{t,h}$	46.3^{t}_{o}	46.9^{t}_{o}	$39.2^{t,h}_{o,c,s}$
WT10G	MAP	19.1	21.9^{t}	$13.3^{t,h}$	21.4^{t}_{o}	$\mathbf{22.9^{t,h}_{o,c}}$	$16.7^{h}_{o,c,s}$
	p@10	27.3	30.4^{t}	$21.6^{t,h}$	30.5^{t}_{o}	$\mathbf{31.6^{t}_{o}}$	$25.3^{h}_{o,c,s}$
	NDCG	30.3	32.7	$21.2^{t,h}$	32.1_{o}	$\mathbf{34.3^{t}_{o,c}}$	$25.4^{h}_{o,c,s}$
GOV2	MAP	29.6	32.1^{t}	$18.0^{t,h}$	30.6^{h}_{o}	$\mathbf{32.2^{t}_{o,c}}$	$20.7^{t,h}_{o,c,s}$
	p@10	53.9	57.3^{t}	$39.4^{t,h}$	56.8_{o}	$\mathbf{57.7^{t}_{o}}$	$44.0^{t,h}_{o,c,s}$
	NDCG	44.8	47.4^{t}	$32.7^{t,h}$	46.9_{o}	$\mathbf{47.9^{t}_{o}}$	$35.7^{t,h}_{o,c,s}$
ClueB	MAP	17.1	18.7^{t}	$14.0^{t,h}$	18.5_{o}	$\mathbf{19.5^{t}_{o}}$	$14.0^{t,h}_{c,s}$
	p@10	22.7	25.9^{t}	23.9	26.7^{t}_{o}	$\mathbf{27.4^{t}}$	24.1
	NDCG	16.5	18.7^{t}	18.3	19.2^{t}	$\mathbf{19.3^{t}}$	17.5
ClueBF	MAP	18.8	$\mathbf{20.5^{t}}$	$14.4^{t,h}$	19.9_{o}	20.3^{t}_{o}	$14.4^{t,h}_{c,s}$
	p@10	33.6	37.9^{t}	29.2^{h}	$\mathbf{38.2^{t}_{o}}$	37.9^{t}_{o}	$30.6^{h}_{c,s}$
	NDCG	24.3	$\mathbf{28.4^{t}}$	22.2^{h}	$\mathbf{28.4^{t}_{o}}$	27.5^{t}_{o}	$22.8^{h}_{c,s}$

the folds. The optimal parameter values for each of the 10 train sets are determined using a simple grid search applied to optimize MAP. The learned parameter values are then used for the queries in the corresponding test fold.

The value of the Dirichlet smoothing parameter, μ, is selected from $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$. The parameter λ, used in HTLM, STLM, F-HT and F-ST, is set to values in $\{0, 0.1, \ldots, 1\}$. The document (τ_d) and query (τ_q) entity-markup confidence level thresholds, used in HT, HTOEnt and F-HT, are set to values in $\{0, 0.1, \ldots, 0.9\}$.

5.2.2 Experimental results

5.2.2.1 Entity-based language models

Table 5.2 presents the performance of the methods that use entity-based language models to instantiate Equation 5.6. Our first observation is that the HT and ST methods outperform the standard term-based language-model retrieval, TermsLM, in all relevant comparisons (6 corpora $\times$ 3 evaluation measures); most improvements are substantial and statistically significant. Furthermore, HT and ST outperform to a substantial and statistically significant degree their special cases which use only entity tokens: HTOEnt and STOEnt, respectively. These findings attest to the merits of using our proposed language models, HTLM and STLM, which integrate term-based and entity-based information.

We also see in Table 5.2 that HT and ST outperform HTCon in most rele-

vant comparisons; most MAP improvements for ST are statistically significant. Recall from Section 5.2.1 that HTCon represents past practice of concept-based representations: accounting uniformly for all entity mentions and attributing equal importance to entity and term tokens. Below we further study the importance of accounting for the confidence level of entity markups, and attributing different weights to term and entity tokens as in HT and ST.

Table 5.2 shows that ST outperforms HT in most relevant comparisons, although rarely to a statistically significant degree. In addition, ST posts more statistically significant improvements over HTCon than HT. We note that HT depends on four free parameters (λ, τ_q, τ_d and μ) while ST depends only on two (λ and μ). Furthermore, the values learned for τ_q and τ_d in HT using the training folds are very low, attesting to the merits of using high recall entity markup. (We revisit this point below.) Overall, these findings attest to the potential merits of using a soft-thresholding approach for the confidence level of entity markups (STLM) with respect to a hard-thresholding approach (HTLM); i.e., accounting for all entity markups in a text and weighing their impact by their confidence levels is superior to accounting, uniformly, for entity markups with a confidence level above a threshold.

Terms vs. entities Figure 5.1 depicts the MAP performance of HT and ST as a function of λ. Low and high values of λ result in more importance attributed to entity-based and term-based information, respectively. For $\lambda = 1$, the two methods amount to TermsLM — i.e., standard term-based language model retrieval. For $\lambda = 0$, the methods use only entity-based information; specifically, HT reduces to HTOEnt and ST reduces to STOEnt.

We see in Figure 5.1 that optimal performance is always attained for $\lambda \notin \{0, 1\}$. This finding echoes those based on Table 5.2. That is, HT and ST outperform TermsLM, and HTOEnt and STOEnt, respectively. Thus, we find that there is much merit in integrating term-based and entity-based information for representing queries and documents.

Figure 5.1 shows that the optimal value of λ for HT is often higher than for ST. This can be attributed to the fact that HTLM, used to represent the query and documents in HT, uses a single confidence-level threshold for entity markups. Thus, potentially valuable information about entities is not utilized. As a result, HT calls for more reliance on term-based information to "compensate" for this potential information loss. In contrast, ST accounts for all entity markups, weighing their impact by their confidence levels. Hence, the "risk" in relying on entity-based information is lower.

To further explore the effect of using a hard threshold for the confidence level of entity markups in HT, we present in Figure 5.2 its MAP performance as a function of τ_q and τ_d — the query and document thresholds, respectively. Recall that low threshold corresponds to high recall markup. Figure 5.2 shows that low values of τ_q and τ_d lead to improved performance. This finding can be attributed to the

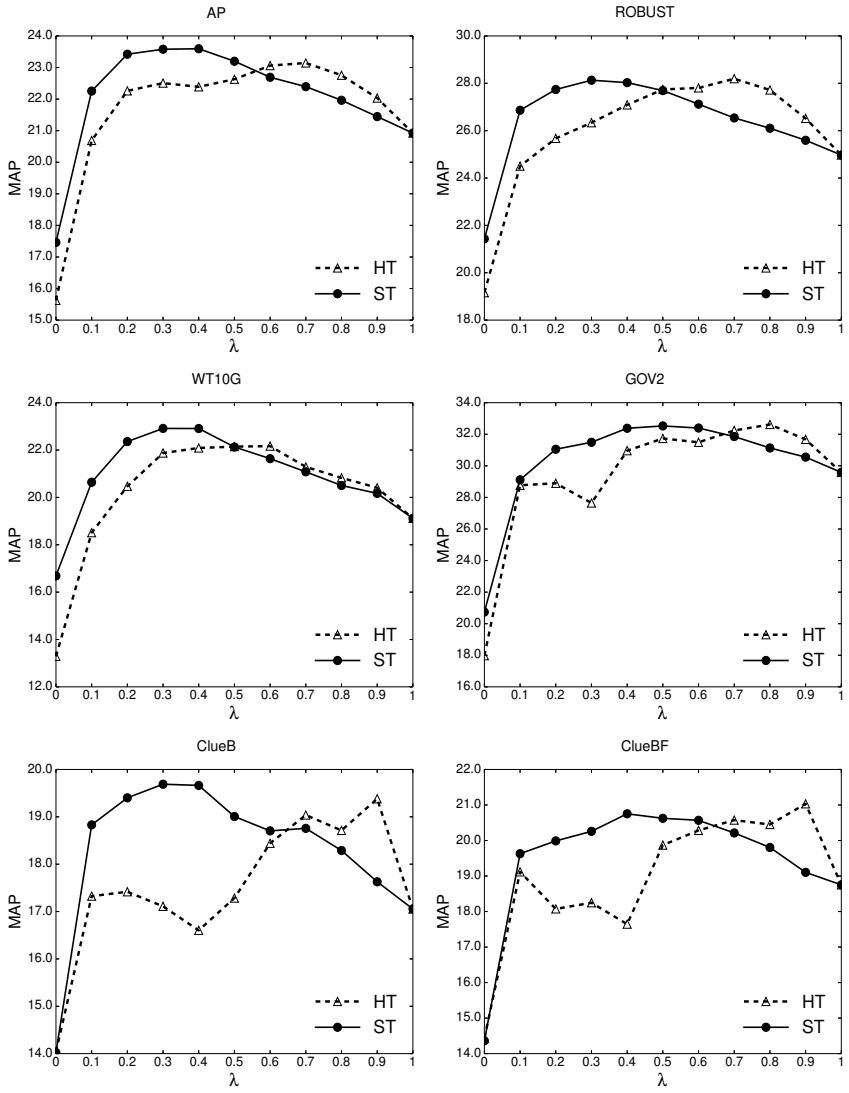

Figure 5.1: The effect of varying λ on the MAP of HT and ST. For $\lambda = 1$, the methods amount to TermsLM (term-based language model retrieval). For $\lambda = 0$, the methods use only entity tokens. The performance is reported for the test folds (i.e., all queries in a dataset) when fixing the value of λ and using cross validation to set the values of all other free parameters. Note: figures are not to the same scale.

fact that increasing the confidence-level threshold amounts to loosing potentially valuable information about appearances of entities in the query and documents. To compensate for the lower precision (i.e., noisier) markup caused by using a low threshold, more weight is put on term-based information as is evident in the relatively high optimal values of λ presented in Figure 5.1. Specifically, we note that the *learned values* of λ, τ_d, and τ_q, averaged over the train folds, for AP,

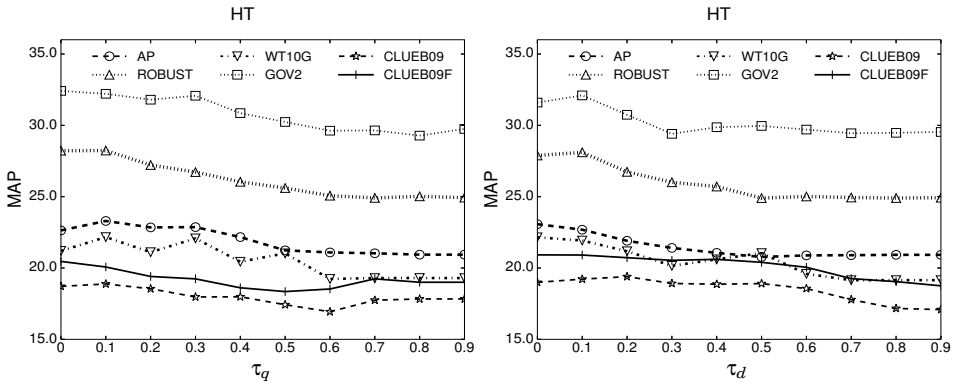

Figure 5.2: The effect of varying τ_q and τ_d on the MAP performance of HT. The values of free parameters, except for that in the x-axis, are set using cross validation as in Figure 5.1.

ROBUST, WT10G, GOV2, ClueB and ClueBF are (0.6, 0.01, 0.11), (0.7,0.1,0.01), (0.55,0.1,0.2), (0.77,0.1,0.01), (0.7,0.15,0), and (0.81, 0.17, 0) respectively; namely, relatively high values of λ and low values of τ_d and τ_q lead to improved performance.

Entity linking Our main evaluation is based on using TagMe for entity linking. In Table 5.3 we compare the retrieval performance when using the entity markups of TagMe and Wikifier. Having Wikifier annotate large-scale collections is a challenging computational task. Thus, we present results only for AP, ROBUST and WT10G. We report MAP and NDCG; the performance patterns for p@10 are the same.

Table 5.3 shows that using ST, our best performing method from above, with Wikifier, results in performance that transcends (often, significantly) that of the standard term-based language model (TermsLM) when using all queries in a dataset (the "All Queries" block). However, the performance of using TagMe is consistently better.

TagMe marks more queries with at least one entity than Wikifier: for AP, ROBUST and WT10G, Wikifier marked no entities in 17, 34 and 26 queries, respectively; TagMe did not mark entities in 0, 1 and 3 queries. (For GOV2 TagMe marked all queries with entities and for ClueB/ClueBF all queries except for one.) Recall that for queries with no marked entities, ST relies only on term-based information.

To refine the comparison of TagMe and Wikifier, we report the performance of ST and STOEnt[5] — the latter relies only on entity tokens — with these two

[5]For queries for which a tool does not mark any entities, no documents are retrieved with STOEnt. Thus, we do not report the performance of STOEnt using all queries as the results are inherently biased in favor of TagMe which marks many more queries with entities than Wikifier.

Table 5.3: Comparing entity-linking tools. Either all queries in a dataset are used ("All Queries"), or only those marked with at least one entity by both TagMe and Wikifier ("Marked Queries"). Bold: best result in a column in a block; 't', 's', 'w' and 'e': statistically significant differences with TermsLM, TagMe-ST, Wikifier-ST and TagMe-STOEnt, respectively.

		AP		ROBUST		WT10G	
		MAP	NDCG	MAP	NDCG	MAP	NDCG
All Queries							
	TermsLM	20.9	40.4	25.0	43.5	19.1	30.3
TagMe	ST	$\mathbf{23.5^t}$	$\mathbf{45.5^t}$	$\mathbf{28.1^t}$	$\mathbf{46.9^t}$	$\mathbf{22.9^t}$	$\mathbf{34.3^t}$
Wikifier	ST	23.3^t	43.6	27.2^t	45.6^t	$19.7^{t,s}$	30.9^s
Marked Queries							
	TermsLM	22.2	41.7	25.4	43.9	21.4	34.2
TagMe	ST	$\mathbf{25.1^t}$	$\mathbf{48.4^t}$	$\mathbf{28.8^t}$	$\mathbf{47.3^t}$	$\mathbf{24.8^t}$	**36.2**
Wikifier	ST	$\mathbf{25.1^t}$	46.2^t	28.0^t	46.4^t	21.9^s	34.0
TagMe	STOEnt	$18.5^{t,s}_w$	41.4^s	$22.9^{t,s}_w$	41.1^s_w	18.1^s	28.1^s
Wikifier	STOEnt	$17.5^{t,s}_w$	39.1^s_w	$19.4^{t,s}_{w,e}$	$34.8^{t,s}_{w,e}$	$12.6^{t,s}_{w,e}$	$21.8^{t,s}_w$

tools over only queries in which both marked at least one entity. As can be seen in the "Marked Queries" block in Table 5.3, TagMe still outperforms Wikifier in almost all relevant comparisons; for STOEnt, several improvements are statistically significant.

TagMe's superiority can be partially attributed to marking more entities (with confidence level > 0) on average than Wikifier: (2.4, 1.8, 2.0) with respect to (1.7, 1.2, 1.0) in queries over AP, ROBUST and WT10G; and, (157.2, 158.7, 207.0) with respect to (58.4, 50.5, 61.7) in documents.

To conclude, our methods are effective with both TagMe and Wikifier. Using TagMe yields better performance that can be partially attributed to higher recall entity markup.

5.2.2.2 The score-based fusion methods

Table 5.4 presents the performance of the F-HT and F-ST methods from Section 5.1.2.1 that perform score fusion of term-only-based and entity-only-based retrieval scores. The performance of TermsLM (term-only language model), HT and ST that integrate term and entity information at the language model level, and that of F-HTCon which is a special case of F-HT (see Section 5.2.1), is presented for reference. We see that F-HT and F-ST substantially outperform TermsLM. (F-ST posts the best performance in most relevant comparisons in Table 5.4.) Both methods also outperform F-HTCon in most relevant comparisons.

In most relevant comparisons, F-HT outperforms HT and F-ST outperforms ST, although most performance differences are not statistically significant. The improvements can be attributed to the fact that F-HT and F-ST use a different smoothing parameter value for terms and entities while HT and ST use a joint one. (See Section 5.1.2.1 for details.)

The potential effectiveness of using different smoothing parameters for term

Table 5.4: Score-based fusion ("F-" methods). Bold: best result in a row; 't', 'h', 's', 'f' and 'c': statistically significant differences with TermsLM, HT, ST, F-HT and F-HTCon, respectively.

		TermsLM	HT	ST	F-HT	F-HTCon	F-ST
AP	MAP	20.9	23.1^{t}	23.5^{t}	23.1^{t}	22.5_{s}	$\mathbf{23.9^{t,h}_{f,c}}$
	p@10	39.1	44.2^{t}	43.8^{t}	$\mathbf{44.5^{t}}$	43.5^{t}	44.2^{t}
	NDCG	40.4	45.3^{t}	45.5^{t}	$\mathbf{46.2^{t}}$	45.1^{t}	45.8^{t}
ROBUST	MAP	25.0	28.1^{t}	28.1^{t}	28.1^{t}	27.7^{t}	$\mathbf{28.4^{t}_{c}}$
	p@10	42.2	45.5^{t}	45.3^{t}	45.7^{t}	45.2^{t}	$\mathbf{46.7^{t}_{s,c}}$
	NDCG	43.5	47.1^{t}	46.9^{t}	47.3^{t}	46.6^{t}	$\mathbf{47.8^{t}_{c}}$
WT10G	MAP	19.1	21.9^{t}	$\mathbf{22.9^{t,h}}$	22.2^{t}	21.6^{t}_{s}	$\mathbf{22.9^{t}_{c}}$
	p@10	27.3	30.4^{t}	31.6^{t}	30.0	30.4^{t}	$\mathbf{31.8^{t}}$
	NDCG	30.3	32.7	$\mathbf{34.3^{t}}$	32.7	33.1	33.7^{t}
GOV2	MAP	29.6	32.1^{t}	32.2^{t}	$\mathbf{33.5^{t,h}_{s}}$	$30.6^{h}_{s,f}$	$33.3^{t,h}_{s,c}$
	p@10	53.9	57.3^{t}	57.7^{t}	$\mathbf{58.6^{t}}$	57.0	58.0^{t}
	NDCG	44.8	47.4^{t}	47.9^{t}	$\mathbf{48.7^{t}}$	46.6	48.2^{t}
ClueB	MAP	17.1	18.7^{t}	19.5^{t}	$19.6^{t,h}$	19.3^{t}	$\mathbf{20.8^{t,h}_{s,f,c}}$
	p@10	22.7	25.9^{t}	27.4^{t}	26.4^{t}	27.5^{t}	$\mathbf{28.8^{t,h}_{f}}$
	NDCG	16.5	18.7^{t}	19.3^{t}	19.1^{t}	19.9^{t}	$\mathbf{20.5^{t,h}_{f}}$
ClueBF	MAP	18.8	20.5^{t}	20.3^{t}	$21.3^{t,h}$	19.7_{f}	$\mathbf{21.8^{t,h}_{s,c}}$
	p@10	33.6	37.9^{t}	37.9^{t}	$\mathbf{39.6^{t}}$	36.5_{f}	39.4^{t}_{c}
	NDCG	24.3	28.4^{t}	27.5^{t}	$\mathbf{29.5^{t}_{s}}$	27.6	29.2^{t}_{s}

and entity tokens stems from the different number of terms and entity markups in a document. The average number of terms in a document for AP, ROBUST, WT10G, GOV2, and ClueB (ClueBF) is 455.4, 474.8, 588.2, 904.7 and 813.6, respectively. The average number of entity markups with a confidence level > 0 is much lower: 157.2, 158.7, 207.0, 291.9 and 307.8.

5.2.2.3 Comparison and integration with SDM

We next compare our entity-based approach with the sequential dependence model (SDM) [111] which scores d by:

$$S_{SDM}(d;q) \stackrel{def}{=} \lambda_S Sim_S(d,q) + \lambda_O Sim_O(d,q) + \lambda_U Sim_U(d,q);$$

the sum of the λ_S, λ_O and λ_U parameters is 1; $Sim_S(d,q)$, $Sim_O(d,q)$ and $Sim_U(d,q)$ are cross-entropy based similarity estimates of the document to the query, utilizing information about occurrences of unigram, ordered bigrams, and un-ordered bigrams, respectively, of q's terms in d; un-ordered bigrams are confined to 8-terms windows in documents.

Using entity tokens in our methods amounts to utilizing information about the occurrences of only *some ordered* variable-length n-grams of query terms in documents — i.e., n-grams which constitute entities. Thus, in contrast to SDM, our methods do not utilize proximity information for query terms which are not in entity markups nor proximity information for unordered n-grams of query terms.

In addition, we study the merit of integrating entity-based information, specifically, our soft-thresholding language model STLM, with SDM. To that end, we

Table 5.5: Comparison and integration with SDM [111]. Bold: the best result in a row. 't', 's', 'f' and 'm' mark statistically significant differences with TermsLM, ST, F-ST and SDM, respectively.

		TermsLM	ST	F-ST	SDM	SDM+STLM
AP	MAP	20.9	23.5^{t}	$\mathbf{23.9^{t}}$	21.6^{s}_{f}	$\mathbf{23.9^{t}_{m}}$
	p@10	39.1	43.8^{t}	$\mathbf{44.2^{t}}$	40.6_{f}	$\mathbf{44.2^{t}_{m}}$
	NDCG	40.4	45.5^{t}	$\mathbf{45.8^{t}}$	42.3_{f}	$\mathbf{45.8^{t}_{m}}$
ROBUST	MAP	25.0	28.1^{t}	$\mathbf{28.4^{t}}$	$25.7^{t,s}_{f}$	28.3^{t}_{m}
	p@10	42.2	45.3^{t}	$\mathbf{46.7^{t,s}}$	43.9^{t}_{f}	$45.7^{t}_{f,m}$
	NDCG	43.5	46.9^{t}	$\mathbf{47.8^{t}}$	$44.8^{t,s}_{f}$	$47.1^{t}_{f,m}$
WT10G	MAP	19.1	22.9^{t}	22.9^{t}	20.2^{s}_{f}	$\mathbf{23.1^{t}_{m}}$
	p@10	27.3	31.6^{t}	$\mathbf{31.8^{t}}$	27.7^{s}_{f}	31.6^{t}_{m}
	NDCG	30.3	$\mathbf{34.3^{t}}$	33.7^{t}	30.7^{s}_{f}	34.0^{t}_{m}
GOV2	MAP	29.6	32.2^{t}	$33.3^{t,s}$	32.1^{t}	$\mathbf{34.7^{t,s}_{f,m}}$
	p@10	53.9	57.7^{t}	58.0^{t}	58.3^{t}	$\mathbf{61.4^{t,s}_{f,m}}$
	NDCG	44.8	47.9^{t}	48.2^{t}	48.4^{t}	$\mathbf{50.6^{t,s}_{f,m}}$
ClueB	MAP	17.1	19.5^{t}	$20.8^{t,s}$	$18.2^{t,s}_{f}$	$\mathbf{21.5^{t,s}_{m}}$
	p@10	22.7	27.4^{t}	28.8^{t}	23.8^{s}_{f}	$\mathbf{30.8^{t,s}_{f,m}}$
	NDCG	16.5	19.3^{t}	20.5^{t}	16.9^{s}_{f}	$\mathbf{21.9^{t,s}_{m}}$
ClueBF	MAP	18.8	20.3^{t}	$21.8^{t,s}$	20.2^{t}_{f}	$\mathbf{22.7^{t,s}_{f,m}}$
	p@10	33.6	37.9^{t}	39.4^{t}	35.8^{t}_{f}	$\mathbf{42.8^{t,s}_{f,m}}$
	NDCG	24.3	27.5^{t}	$29.2^{t,s}$	25.9^{t}_{f}	$\mathbf{32.2^{t,s}_{f,m}}$

augment the SDM scoring function with an entity-based document-query similarity estimate, $Sim_E(d,q)$. For this estimate, we use the score assigned to d by the STOEnt method; i.e., we use an entity-only language model since term-based information is accounted for in $Sim_S(d,q)$. The resultant method, **SDM+STLM**, scores d by ($\lambda_S + \lambda_O + \lambda_U + \lambda_E = 1$):

$$S_{SDM+STLM}(d;q) \stackrel{def}{=} \lambda_S Sim_S(d,q) + \lambda_O Sim_O(d,q) + \lambda_U Sim_U(d,q) + \lambda_E Sim_E(d,q).$$

SDM+STLM can be viewed as a novel instantiation of a weighted dependence model (WSDM) [19] with a novel concept type (i.e., entity). If $\lambda_O = \lambda_U = 0$, SDM+STLM amounts to our F-ST method (see Section 5.1.2.1).

All free parameters of SDM and SDM+STLM: λ_S, λ_O, λ_U, λ_E and the Dirichlet smoothing parameter, μ, are set using cross validation as described in Section 5.2.1; λ_S, λ_O, λ_U, and λ_E are selected from $\{0, 0.1, \ldots, 1\}$ and μ is set to values in $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$.

Table 5.5 shows that ST and F-ST outperform SDM, often statistically significantly, in most relevant comparisons (6 corpora $\times$ 3 evaluation measures). This implies that using variable length n-grams which potentially bear semantic meaning (entities) can yield better performance than using ordered and unordered bigrams which do not necessarily have semantic meaning. Recall that in contrast to SDM, ST and F-ST do not account for proximities between terms which do not constitute entities and for unordered bigrams.

Table 5.6: Robustness analysis. Number of queries for which ST hurts (-) and improves (+) AP performance with respect to TermsLM and SDM.

	AP		ROBUST		WT10G		GOV2		ClueB		ClueBF	
	-	+	-	+	-	+	-	+	-	+	-	+
ST vs. TermsLM	38	61	75	173	31	63	50	99	54	137	75	112
ST vs. SDM	35	64	87	161	33	60	74	75	79	112	89	97

In most relevant comparisons, SDM+STLM outperforms SDM and ST (which utilizes STLM) and is as effective as, and often posts statistically significant improvements over, F-ST — its special case that fuses unigram term-only and entity-only retrieval scores. The few cases where F-ST outperforms SDM+STLM could be attributed to potential over-fitting effects due to the high number of free parameters of SDM+STLM and the relatively low number of queries.

We also found that effective weights assigned to entity-only similarities in SDM+STLM (λ_E) are much higher than those assigned to ordered (λ_O) and un-ordered (λ_U) bigram term-based similarities. Furthermore, effective values of λ_O and λ_U are lower and higher, respectively, for SDM+STLM than for SDM. These findings further attest to the merits of using entity-based similarities with respect to (ordered and un-ordered) bigram similarities, and show that un-ordered bigram, in contrast to ordered bigram, similarities could be complementary to entity-based similarities

5.2.2.4 Further analysis

We now turn to further analyze merits, and shortcomings, of using entity-based query and document representations. To that end, we focus on the ST method that utilizes STLM.

Table 5.6 presents performance robustness analysis: the number of queries for which ST improves or hurts average precision (AP) over TermsLM and SDM. In both cases, ST improves AP for more queries than it hurts; naturally, the differences with SDM are smaller than those with TermsLM.

One advantage of STLM is that it represents the query and documents using entities which constitute variable length n-grams with semantic meaning. A case in point, query #41 in ClueWeb, "orange county convention center", refers to the primary public convention center for the Central Florida region. TermsLM, SDM and ST ranked the Web home page for this entity second. However, at the third rank in the lists retrieved by TermsLM and SDM appears a Wikipedia page titled "list of convention and exhibition centers", which is not specific to the entity of concern. The average precision (AP) of TermsLM, SDM and ST for the query in the ClueB dataset was 9, 13, and 30, respectively, attesting to the merit of the correct identification of the entity in the query and its utilization by ST.

The ST method can suffer from incorrect entity identification in queries. For example, query #407 in ROBUST, "poaching, wildlife preserves", targets informa-

tion about the impact of poaching on the world's various wildlife preserves. The entities identified by TagMe are "poaching", "wildlife" and "preserves"; the latter refers to fruit preserves instead of nature preserves. Such erroneous entity identification can be attributed to the little context short queries provide. Consequently, the AP of ST for this query is only 8 while that of TermsLM and SDM is 31.4 and 30.0, respectively.

5.2.3 Using entity-based language models in additional retrieval paradigms

We next explore the effectiveness of using our entity-based language models in two additional retrieval paradigms: cluster-based document retrieval and query expansion.

5.2.3.1 Cluster-based document retrieval

Let D_{init} denote the list of top-n documents retrieved by TermsLM (standard language-model-based retrieval). Following common practice in work on cluster-based document retrieval [101, 88], we re-rank D_{init} using information induced from nearest-neighbor clusters of documents in D_{init}.

We use $Sim(x, y) \stackrel{def}{=} exp(-CE\left(\theta_x^{MLE} \mid\mid \theta_y^{Dir}\right))$ to measure the similarity between texts x and y [88]; θ_x^{MLE} is an unsmoothed MLE induced from x and θ_y^{Dir} is a Dirichlet smoothed language model induced from y. Each document d ($\in D_{init}$) and the $k-1$ documents d' ($d' \neq d$) in D_{init} that yield the highest $Sim(d, d')$ constitute a cluster.

We rank the (overlapping) clusters c, each contains k documents, by: $\sqrt[k]{\prod_{d \in c} Sim(q, d)}$ [101]. This is a highly effective simple cluster ranking method [88]. To induce document ranking, each cluster is replaced with its constituent documents omitting repeats; documents in a cluster are ordered by their query similarity: $Sim(q, d)$.

The document (re-)ranking procedure just described relies on the choice of the document language models used to induce clusters (i.e., in $Sim(d, d')$) and the choice of document and query language models used to induce document-query similarities ($Sim(q, d)$); the latter are used for ranking both clusters and documents within the clusters. We use **C-Term-Term** to denote the standard method that uses term-only language models for inducing clusters and document-query similarities [101, 88]. The **C-Term-Ent** method utilizes the same clusters used by C-Term-Term, but uses our entity-based language model, STLM, for inducing document-query similarities to rank clusters and documents in them. In the **C-Ent-Ent** method, STLM is used to both create clusters and induce document-query similarities. As a reference comparison, we re-rank D_{init} using the ST method that uses STLM but does not utilize clusters.

As the main goal of cluster-based re-ranking is improving precision at top ranks [101, 88], we report p@10 and NDCG@10 (NDCG). Free-parameter values are

Table 5.7: Cluster-based document re-ranking. Bold: the best result in a row; 't', 's', '$*$' and 'ψ' mark statistically significant differences with TermsLM, ST, C-Term-Term and C-Term-Ent, respectively.

		TermsLM	ST	C-Term-Term	C-Term-Ent	C-Ent-Ent
AP	p@10	39.6	42.5	43.2^{t}	44.3^{t}	$\mathbf{46.5^{t,s}}$
	NDCG	40.8	44.8^{t}	44.2^{t}	44.9	$\mathbf{46.8^{t}}$
ROBUST	p@10	42.2	44.3^{t}	43.1	46.0^{t}_{*}	$\mathbf{47.7^{t,s}_{*,\psi}}$
	NDCG	43.5	45.5^{t}	44.2	47.5^{t}_{*}	$\mathbf{49.1^{t,s}_{*,\psi}}$
WT10G	p@10	28.6	30.6	30.2	$33.7^{t,s}_{*}$	$\mathbf{34.8^{t,s}_{*}}$
	NDCG	31.2	33.4	32.1	35.4^{t}_{*}	$\mathbf{36.3^{t,s}_{*}}$
GOV2	p@10	53.4	57.0^{t}	55.1	$\mathbf{58.3^{t}}$	57.9^{t}
	NDCG	45.0	46.8	45.8	$\mathbf{48.9^{t}}$	47.8^{t}
ClueB	p@10	23.7	27.1^{t}	23.7	$\mathbf{33.0^{t,s}_{*}}$	$31.5^{t,s}_{*}$
	NDCG	17.2	19.1	17.2	$\mathbf{24.9^{t,s}_{*}}$	$22.9^{t,s}_{*}$
ClueBF	p@10	32.1	36.9^{t}	31.2^{s}	38.5^{t}_{*}	$\mathbf{39.0^{t}_{*}}$
	NDCG	22.9	27.8^{t}	23.1^{s}	$\mathbf{30.3^{t}_{*}}$	29.6^{t}_{*}

set using cross validation; NDCG is the optimization criterion. Specifically, n is selected from $\{50, 100\}$; k is in $\{5, 10\}$; and, λ (used in STLM) is in $\{0, 0.1, \ldots, 1\}$; the Dirichlet smoothing parameter is set to 1000. Table 5.7 presents the results.

We see that all cluster-based methods (denoted "C-X-Y") almost always outperform the initial term-based document ranking, TermsLM. C-Term-Ent substantially outperforms C-Term-Term. This attests to the merits of using STLM for inducing cluster ranking and within cluster document ranking. In most relevant comparisons, C-Ent-Ent outperforms (and is never statistically significantly outperformed by) C-Term-Ent, attesting to the potential merits of using entity-based information to also create clusters. However, only two improvements are statistically significant.

Finally, Table 5.7 shows that in almost all relevant comparisons, ST outperforms TermsLM (often, statistically significantly) and C-Term-Term and is outperformed by C-Term-Ent and C-Ent-Ent. This shows that while there is merit in using STLM for direct ranking of documents as shown in Section 5.2.2.1, the performance can be further improved by using STLM for cluster-based document ranking.

5.2.3.2 Query expansion

There is much work on expanding queries with terms and entities using entity-based information [160, 109, 120, 23, 42, 99, 156, 95, 103]. In contrast, our entity-based language models, when induced from the query, utilize only query terms and entities marked in the query. Hence, we study the effectiveness of using our language models to perform query expansion.

We use the relevance model (RM3) [1] as a basis for instantiating expanded

Table 5.8: Query expansion. Bold: the best result in a row. 't', 's', 'r', 'w', 'm' and 'n' mark statistically significant differences with TermsLM, ST, RM3, WikiRM, SDM-RM and RMST, respectively.

		TermsLM	ST	RM3	WikiRM	SDM-RM	RMST	RMST-ST
AP	MAP	20.9	23.5^{t}	24.1^{t}	24.0^{t}	24.9^{t}	24.6^{t}	$\mathbf{27.4^{t,s,r}_{w,m,n}}$
	p@10	39.1	43.8^{t}	42.5^{t}	46.2^{t}	43.9^{t}	44.8^{t}	$\mathbf{46.8^{t,r}}$
	NDCG	40.4	45.5^{t}	43.2	$\mathbf{48.2^{t,r}}$	45.6^{t}	45.0^{t}	$47.4^{t,r}$
ROBUST	MAP	25.0	28.1^{t}	28.3^{t}	27.8^{t}	28.4^{t}	29.0^{t}	$\mathbf{30.5^{t,s,r}_{w,m,n}}$
	p@10	42.2	45.3^{t}	43.6	44.6^{t}	43.2	$45.9^{t,r}_{m}$	$\mathbf{47.1^{t,s,r}_{w,m}}$
	NDCG	43.5	46.9^{t}	43.8^{s}	$46.1^{t,r}$	43.6^{s}_{w}	$46.5^{t,r}_{m}$	$\mathbf{47.2^{t,r}_{m}}$
WT10G	MAP	19.1	$\mathbf{22.9^{t}}$	19.6^{s}	$21.9^{t,r}$	20.0^{s}	$22.7^{t,r}_{m}$	$22.8^{t,r}_{m}$
	p@10	27.3	31.6^{t}	28.0^{s}	$\mathbf{34.2^{t,r}}$	28.6_{w}	$31.7^{t,r}_{m}$	31.1^{t}
	NDCG	30.3	$\mathbf{34.3^{t}}$	30.1^{s}	$\mathbf{34.3^{t,r}}$	30.5^{s}_{w}	32.9	31.8^{s}
GOV2	MAP	29.6	32.2^{t}	32.4^{t}	32.1^{t}	$\mathbf{33.7^{t}_{w}}$	33.1^{t}	$\mathbf{33.7^{t,s}}$
	p@10	53.9	57.7^{t}	58.1^{t}	$\mathbf{60.1^{t}}$	58.0^{t}	59.6^{t}	58.5^{t}
	NDCG	44.8	47.9^{t}	48.0^{t}	$\mathbf{50.6^{t}}$	47.6	49.4^{t}	48.8^{t}
ClueB	MAP	17.1	19.5^{t}	19.3^{t}	$21.9^{t,s,r}$	$20.9^{t,r}$	$20.7^{t,s,r}$	$\mathbf{22.1^{t,s,r}_{n}}$
	p@10	22.7	27.4^{t}	30.6^{t}	$\mathbf{35.3^{t,s,r}}$	$32.2^{t,s}_{w}$	$32.2^{t,s}_{w}$	$34.9^{t,s,r}_{n}$
	NDCG	16.5	19.3^{t}	$22.6^{t,s}$	$26.1^{t,s,r}$	$24.3^{t,s}$	$25.1^{t,s,r}$	$\mathbf{27.1^{t,s,r}_{n}}$
ClueBF	MAP	18.8	20.3^{t}	20.4^{t}	21.0^{t}	$21.8^{t,s,r}$	20.8^{t}	$\mathbf{21.9^{t,s}_{n}}$
	p@10	33.6	37.9^{t}	37.9^{t}	38.5^{t}	$\mathbf{39.7^{t,r}}$	38.2^{t}	38.4^{t}
	NDCG	24.3	27.5^{t}	28.1^{t}	28.2^{t}	$29.8^{t,r}$	28.5^{t}	$\mathbf{30.3^{t,s}}$

query forms. The probability assigned to *token* t by a relevance model RM is:

$$RM(t) \stackrel{def}{=} \alpha\theta_q^{MLE}(t) + (1-\alpha)\sum_{d\in L}\theta_d^{Dir}(t)\frac{S(d;q)}{\sum_{d'\in L}S(d';q)}; \tag{5.8}$$

α is a free parameter; L is a list of top-retrieved documents used to construct RM; $S(d;q)$ is d's score. Due to computational considerations, as in work on entity-based query expansion [42, 156] we use RM to re-rank an initially retrieved document list; $CE\left(RM \mid\mid \theta_d^{Dir}\right)$ serves for re-ranking.

Using only terms as tokens, and applying standard language-model-based retrieval (TermsLM) over the corpus to create L, yields the standard **RM3** [1]. Creating L by applying TermsLM over Wikipedia results in **WikiRM** [160], an external corpus expansion approach also used in [42, 156]. RM3 and WikiRM re-rank a document list retrieved by TermsLM. (WikiRM is the only model where the list from which RM is constructed, L, is not a sub-set of the list to be re-ranked.) In both methods, $S(d;q) \stackrel{def}{=} \exp(-CE\left(\theta_q^{MLE} \mid\mid \theta_d^{Dir}\right))$.

The **SDM-RM** model [42] is constructed from, and used to re-rank, lists retrieved by the sequential dependence model (SDM) [111]. θ_d^{Dir}, and the resultant relevance model constructed by setting $\alpha = 0$ in Equation 5.8, are term-based unigram language models; $S(d;q)$ is the exponent of the score assigned to d by SDM. Re-ranking is performed by linear interpolation of the SDM score assigned to d and $-CE\left(RM \mid\mid \theta_d^{Dir}\right)$, using a parameter α. SDM-RM is, in fact, the highly effective Latent Concept Expansion method [112] without IDF-based weighting of expansion terms.

The next two relevance models, defined over $\mathcal{T}$ (the term-entity token space

from Equation 5.1), are novel to this study. They utilize our STLM language model which integrates terms and entities at the language model level. **RMST** is inspired by methods proposed by Dalton et al. [42][6] by the virtue of using both terms and entities for query expansion. θ_q^{MLE} and θ_d^{Dir} are our STLM language models. $S(d;q) \stackrel{def}{=} \exp(-CE\left(\theta_q^{MLE} \mid\mid \theta_d^{Dir}\right))$. The TermsLM method is applied over the corpus to create the initial list to be re-ranked (cf. [156]) and from which L is derived.

RMST-ST is constructed as RMST using STLM. The difference is that our entity-based ST method, rather than TermsLM, is used to create the initial list to be re-ranked and from which L is derived. The formal ease of using STLM in the relevance model (Equation 5.8), yielding RMST and RMST-ST, attests to the merits of using a single language model defined over terms and entities with respect to the alternative score-based fusion approach from Section 5.1.2.1.

The free parameters of all methods are set using cross validation. The number of expansion terms (i.e., those assigned the highest probability by RM), the number of documents in L, and α are set to values in $\{10, 30, 50, 100\}$, $\{50, 100\}$ and $\{0, 0.1, \ldots, 1\}$, respectively. (Only for WikiRM, the number of documents in L is selected from $\{1, 5, 10, 30, 50, 100\}$ following [160].) All lists that are re-ranked contain 1000 documents. The values of the free parameters of ST and SDM are selected from the ranges specified in Section 5.2.1. The Dirichlet smoothing parameter, μ, is selected from $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$; for relevance model construction (Equation 5.8) the value 0 is also used (yielding unsmoothed MLE). To reduce the number of free-parameter values configurations, we use the same value of μ for creating L, for re-ranking and for constructing the relevance model, unless 0 is used for relevance model construction.

Table 5.8 presents the performance. Our ST method, which does not perform query expansion, is competitive with the term-based relevance model (RM3). We also see that RMST is an effective expansion method which often outperforms RM3 and SDM-RM. This finding echoes those from past work [42, 156] about the merits of using both terms and entities for query expansion. The best performing method in most relevant comparisons is RMST-ST which uses STLM to (i) create an effective initial list for re-ranking; (ii) create an effective list, L, for relevance model construction; and, (iii) induce ranking using the entity-based relevance model as in RMST. We conclude that our STLM language model can play different important roles in query expansion.

Table 5.8 shows that expansion using Wikipedia as an external corpus (WikiRM) is effective. Our RMST and RMST-ST expansion methods (as well as ST) utilize entity tokens marked by TagMe (i.e., Wikipedia concepts), but do no use the text on their Wikipedia pages in contrast to WikiRM. Thus, integrating WikiRM with our methods, e.g., using score-based integration [42], is interesting future direction.

[6]Various expansion methods, which utilize also auxiliary information about entities from the entity repository, were integrated in [42]. We do not use such auxiliary information.

Chapter 6
Inducing Query Models Using Inter-Entity Similarities

In the ad hoc document retrieval task, a user information need is expressed by a query. The query is usually short and therefore a more informative representation is required for an effective document relevance estimation. There is much work on inducing query models; for example, the relevance model (Lavrenko and Croft [93]) is a highly effective query model. Having a query model constructed by utilizing *some* estimation method, one can estimate the document relevance by comparing the query and the document models.

In Chapter 5 we suggested novel retrieval methods utilizing *surface level* entity-based query and document representations. In this chapter we turn to explore methods utilizing *entity associated information* for inducing novel entity-based query models. Specifically, entity associated documents, Wikipedia links and co-occurrence statistics are used for estimating semantic similarities between pairs of entities in some pre-defined entity set. These estimates are then used for inducing *inter-entity similarity-based query models*. In Section 6.1 we formally define a retrieval framework utilizing these models.

To evaluate whether inter-entity similarities are potentially useful for inducing effective query models, in Section 6.2 we suggest a "second-order cluster hypothesis" for entities: *closely associated entities tend to be relevant to the same requests*. The underlying assumption, different from the well known cluster hypothesis for document retrieval [144], is that the type of retrieved item (document) can be different from the type of the item for which the hypothesis is stated (entities); hence, we term our hypothesis "second-order".

Testing the second-order cluster hypothesis requires relevance estimates for entities. In Section 6.2.1 we suggest an operational method for generating such estimates which relies on the use of evaluated entities to induce query models. Vorhees' nearest-neighbor cluster hypothesis test [146] is suggested as a test for our proposed hypothesis.

In Section 6.3 we present an empirical evaluation of the second-order cluster hypothesis which consists of various experimental settings created by varying the datasets, baselines and similarity measures used for estimating inter-entity similarities. We show that the second-order cluster hypothesis holds to a substantial extent for all these settings. Our findings imply that inter-entity similarities can potentially be utilized for identifying relevant entities.

Encouraged by findings about the second-order cluster hypothesis we propose several operational methods for inducing query models by utilizing inter-entity similarities. The first method utilizes similarities between entities in some pre-defined entity set and the query constituent entities. The entities most similar to the query entities are assigned high query model probabilities. Two additional methods utilize similarities between entities constituting some pre-defined entity

set for inducing entity-based query models. The first of these methods assigns high query model probabilities to central entities. The second method ranks entity clusters, composed of entities highly similar to each other, according to the similarity of their constituting entities to the query. Highest ranked entity clusters are used for inducing the query model. The description of all methods is provided in Section 6.4.

The inter-entity similarity-based query models we propose are utilized for retrieval by several retrieval methods. Empirical evaluation of these methods which consists of various experimental settings is presented in Section 6.5. The empirical findings demonstrate the merits of using inter-entity similarities for retrieval. We show that retrieval methods utilizing our proposed inter-entity similarity-based query models improve retrieval effectiveness with respect to a few effective baselines. In addition, we perform oracle experiments which demonstrate the considerable potential of using clusters of similar entities to induce effective entity-based query models.

6.1 Retrieval Framework

In what follows we assume that *some* retrieval method was employed for ranking documents in corpus D in response to a query q. Let D_{init} be a list composed of documents most highly ranked in the initially retrieved list. We present retrieval methods that re-rank documents in D_{init} so as to improve retrieval performance. We will use d to denote a document in D_{init}.

The methods we suggest utilize information about entities marked in the query and in documents. As described in Chapter 5, the entities are marked using *some* entity linking tool. The entity markup of a term sequence is composed of an entity ID and a confidence level in $[0, 1]$. The confidence level reflects the likelihood that the term sequence corresponds to the entity.

Term-only and entity-only *document* language models, θ_d^{term} and θ_d^{ent}, respectively, are induced for each document in D_{init} by utilizing the soft confidence-level thresholding language models (STLM) presented in Section 5.1.1.2. To induce term-only language model we set λ, which is the free parameter controlling the relative importance attributed to term and entity tokens, to $\lambda = 1$. (See Equation 5.5 on page 36). Then, STLM reduces to a standard unigram term-based language model. Setting $\lambda = 0$ results in an entity-only language model. Following common practice [170], we use Dirichlet smoothed document language models (Equation 5.3 on page 35).

In the following sections we propose methods for inducing term-only and entity-only *query* models, θ_q^{term} and θ_q^{ent}, respectively. We assign document d the following retrieval score with respect to q:

$$S(q; d) \stackrel{def}{=} \lambda CE\left(\theta_q^{term} \,||\, \theta_d^{term}\right) + (1 - \lambda) CE\left(\theta_q^{ent} \,||\, \theta_d^{ent}\right); \tag{6.1}$$

$CE(\cdot \,||\, \cdot)$ is the cross entropy measure, higher values correspond to decreased similarity. λ is a free parameter.

Equation 6.1 is equivalent to Equation 5.7 in Section 5.1.2.1. In this chapter we follow the approach of integrating term and entity information at the *retrieval level score*, which was shown to be highly effective. Instantiating Equation 6.1 with unsmoothed maximum likelihood estimates of the term-only and entity-only query soft confidence-level thresholding language models results in the $F-ST$ retrieval method presented in Chapter 5. In Section 6.4 we describe methods for inducing more informative query models.

6.1.1 Relevance model estimation

The relevance model [93] is a highly effective query model which we use for two different purposes. First, retrieval methods utilizing the relevance model as a query model serve for reference comparison in our empirical evaluation (see Section 6.5). Second, the entities that were assigned high probabilities by the relevance model are used in cluster hypothesis tests (see Section 6.2.1) and also for inducing similarity-based query models. In the following, we describe the induction of term-only and entity-only relevance models.

Documents in D_{init} are re-ranked using the entity-only or term-only STLM language models of the query and the documents in the list. The document score is calculated by instantiating Equation 6.1 with unsmoothed maximum likelihood estimates for the entity-only and term-only query models and with entity-only and term-only Dirichlet smoothed document language models. Setting $\lambda = 1$ results in a term-only document score and $\lambda = 0$ in an entity-only document score.

Since the λ parameter value determines which token type is utilized, we use λ to denote token type dependency. Specifically, the document score is denoted by $S_\lambda(q;d)$ and the re-ranked list is denoted by D_λ; $S_{\lambda=1}(q;d)$ and $S_{\lambda=0}(q;d)$ are the term-only and entity-only document scores, respectively. $D_{\lambda=1}$ and $D_{\lambda=0}$ are the term-based and entity-based re-ranked lists, respectively.

The relevance model $RM1$ is defined by:

$$p(t|RM1) \stackrel{def}{=} \sum_{d \in D_\lambda} p(t|\theta_d^\lambda) \frac{\exp(-S_\lambda(d;q))}{\sum_{d' \in D_\lambda} \exp(-S_\lambda(d;q))}; \tag{6.2}$$

t is a token which can be either an entity or a term. D_λ is the re-ranked document list described above. $p(t|\theta_d^\lambda)$ is the probability assigned to a token t by the document d STLM model; setting $\lambda = 0$ results in an entity-only STLM language model, setting $\lambda = 1$ results in a term-only STLM language model.

$p(e|RM1)$ and $p(t|RM1)$ are the induced entity-only and term-only relevance model distributions, respectively.

6.2 The Second-Order Cluster Hypothesis

To evaluate whether inter-entity similarities are potentially useful for inducing effective query models we turn to explore the cluster hypothesis which is a fundamental concept in retrieval: "closely associated documents tend to be relevant to the same requests" [144]. The original hypothesis is stated for documents, which traditionally serve as *retrieval units*. However, in Chapter 3 we showed that the hypothesis holds for entities, which also serve as retrieval units.

As demonstrated throughout this work, entities play double role in retrieval. On one hand, they serve as retrieval units in the task of entity retrieval [12, 47, 107, 127, 128]. On the other hand, they serve as *information units* utilized by various retrieval methods for addressing the task of ad hoc document retrieval [42, 129, 156].

We suggest a novel view of the cluster hypothesis which is based on the assumption that the type of retrieved item can be different from the type of the item for which the hypothesis is stated. The hypothesis, which we name "second-order cluster hypothesis", is: "closely associated *information units* tend to be relevant to the same requests". Information unit is defined as a meaningful unit of information who's semantic relatedness with another information unit can be estimated, e.g., entities.

In the following we assume that the information unit is an entity marked in the corpus D by an entity linking tool. The retrieval unit is a document, as in the original hypothesis [144]. The second-order cluster hypothesis can also be defined by setting terms as the information units or entities as retrieval units. We leave this research direction for future work.

Several cluster hypothesis tests were proposed [52, 78, 128, 139, 146], based on the assumption that relevance judgments of retrieval units such as documents or entities are provided. To devise a well defined second-order cluster hypothesis test, we need to properly define entity relevance as well as to suggest operational entity relevance estimation method. We now turn to suggest such method.

6.2.1 Estimating entity relevance

An entity is defined relevant with respect to the query if it is related to the user's information need. This definition is very hard to operationalize. In the following, we suggest to estimate entity relevance by fixing the retrieval method which utilizes entity-based information.

The retrieval method that we use is based on the retrieval framework presented in Section 6.1. Documents in D_{init} are re-ranked by utilizing entity-only and term-only query and documents STLM language models. The re-ranked list is set as a baseline ranking, denoted by D_{base}. To estimate the relevance of some entity e, we use the entity for inducing a query model that is used, together with the original query, for re-ranking D_{base}. If using the entity improves retrieval effectiveness with

respect to that of D_{base}, it is defined as relevant. Otherwise, it is defined non-relevant. This retrieval method can be viewed as an entity-based query expansion.

We determine entity relevance for entities constituting a close entity set, S_r. We define S_r to be a set composed of the c entities assigned the highest probabilities by the entity-only relevance model, induced as described in Section 6.1.1. Since the relevance model is considered as a highly effective query model we assume that S_r is composed of entities that are potentially relevant with respect to the query[1].

To formally define the entity relevance estimation method, we modify the score assigned to each document d in D_{init} with respect to the query q, suggested in Section 6.1 (see Equation 6.1), to consider two entity-based query models. The document score is defined as:

$$S(q;d) \stackrel{def}{=} \lambda_1 CE\left(\theta_q^{term} \mid\mid \theta_d^{term}\right) + \lambda_2 CE\left(\theta_q^{ent} \mid\mid \theta_d^{ent}\right) + \lambda_3 CE\left(\theta_q^{exp} \mid\mid \theta_d^{ent}\right); \quad (6.3)$$

$CE\left(\cdot \mid\mid \cdot\right)$ is the cross entropy measure. θ_d^{term} and θ_d^{ent} are the term-only and entity-only Dirichlet smoothed document STLM language models induced as described in Section 6.1. θ_q^{term} and θ_q^{ent} are the unsmoothed maximum likelihood estimates of the term-only and entity-only query STLM language models. θ_q^{exp} is the unsmoothed maximum likelihood estimate of an entity-only query, composed of the entity who's relevance we want to estimate. λ_1, λ_2 and λ_3 are free parameters, $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

To create the D_{base} list, which serve for entity relevance estimation, the initially retrieved list, D_{init}, is re-ranked using three different methods; each method utilizes a different configuration of λ_i parameters, $i \in \{1, 2, 3\}$ (see Equation 6.3). In all three methods, no expansion is used, i.e., $\lambda_3 = 0$. Setting $\lambda_1 = 0, \lambda_3 = 0$ results in the first retrieval method, referred to as $\mathbf{E_Q}$, which utilizes *entity-only* queries for re-ranking D_{init}. Setting $\lambda_2 = 0, \lambda_3 = 0$ results in the second method, referred to as $\mathbf{T_Q}$, which utilizes *term-only* queries for re-ranking D_{init}. The third method, $\mathbf{T_Q E_Q}$, utilizes *both entity and term queries* for re-ranking the documents, i.e., we set $\lambda_1, \lambda_2 \neq 0, \lambda_3 = 0$. We note that T_Q is the standard term-based unigram language model retrieval [92]. E_Q and $T_Q E_Q$ were proposed and evaluated in Chapter 5 (referred to as STOEnt and F-ST). We use D_{base}^R to denote a list created by utilizing retrieval method R; i.e., $D_{base}^{E_Q}$, $D_{base}^{T_Q}$ and $D_{base}^{T_Q E_Q}$ are lists created by utilizing the E_Q, T_Q and $T_Q E_Q$ retrieval methods, respectively.

The relevance of entity e is determined by utilizing the evaluated entity-based document score, $CE\left(\theta_q^{exp} \mid\mid \theta_d^{ent}\right)$, for re-ranking D_{base}^R. Specifically, the parameters λ_1 and λ_2 are set as in the method used to create D_{base}^R and λ_3 is set be non-zero, $\lambda_3 > 0$. The AP scores of D_{base}^R and of the re-ranked list are calculated and compared. If the change in AP score is positive, i.e., utilizing entity e to

[1]Additional entity sets can be utilized for estimating entity relevance, for example, the set of all entities marked in the initial document list, D_{init}, with any confidence level. However, such set is very large and potentially noisy. We therefore chose to experiment with the relevance model-based entity set.

induce a query model improves the document ranking, the entity is defined relevant. Otherwise, it is defined non-relevant.

6.2.2 Testing the second-order cluster hypothesis

Voorhees' nearest-neighbor cluster hypothesis test [146] is used for testing the second-order cluster hypothesis. The test is performed for a query q with respect to a set of potentially relevant entities S_r, given the relevance estimates of entities in S_r with respect to the query. Entity relevance is estimated as suggested in Section 6.2.1. We use three different lists, $D_{base}^{E_Q}$, $D_{base}^{T_Q}$ and $D_{base}^{T_Q E_Q}$ for estimating the entity relevance.

For each relevant entity we count the number of relevant entities in S_r that are among the k nearest neighbors of the entity. Similarities between entities in the list are estimated by utilizing *some* inter-entity similarity measure[2]. The sum of counts over all the relevant entities found for all tested queries is divided by the total number of relevant entities to calculate the test result.

6.3 Evaluation of the Second-Order Cluster Hypothesis Test

We now turn to study the cluster hypothesis using the test that was devised in Section 6.2.

6.3.1 Experimental setup

Experiments for testing the second-order cluster hypothesis as well as additional methods proposed in this chapter were conducted using the TREC datasets specified in Table 6.1. ROBUST is mostly composed of news articles. WT10G is a small, noisy, Web collection. GOV2 is a larger Web collection composed of high quality pages crawled from the .gov domain. ClueBF is the English part of the Category B of the ClueWeb 2009 Web collection. It was created by retrieving an initial result list D_{init} of 10000 documents from ClueWeb 2009 (the initial retrieval method utilized for all collections is described below) and filtering from its rankings suspected spam documents: those assigned a score below 50 by Waterloo's spam classifier [37].

Data processing Titles of TREC topics served for queries. Tokenization and Porter stemming were applied using the Lucene toolkit (lucene.apache.org) which was used for experiments. Stopwords on the INQUERY list were removed from queries but not from documents.

The **TagMe** entity-linking tool (`tagme.di.unipi.it`) is used to annotate non-stemmed and non-stopped queries and documents utilizing Wikipedia (a July 2014

[2]The similarity measures we use are detailed in Section 6.3.

Table 6.1: TREC data used for experiments.

corpus	# of docs	data	queries
ROBUST	528,155	Disks 4-5 (-CR)	301 − 450, 601 − 700
WT10G	1,692,096	WT10g	451 − 550
GOV2	25,205,179	GOV2	701 − 850
ClueBF	50,220,423	ClueWeb09 (Cat. B)	1 − 200

dump) as the entity repository. The annotations process is identical to that described in Section 5.2.1.

Baselines As a reference comparison to the test result we use the mean, over queries and relevant entities, precision of the $N-1$ items in the set of potentially relevant entities, S_r, N is the set size. This baseline, denoted *rand*, is an estimate of the probability of randomly selecting a relevant entity as a neighbor of the relevant entity, rather than by utilizing inter-entity similarities.

Similarity measures We use a variety of inter-entity similarity measures that utilize different types of entity-associated information. The texts of two entities Wikipedia pages are compared by the **COS** measure which was proposed in past for estimating inter-entity similarities utilized for document retrieval [103]. Specifically, 200 terms having the highest TF.IDF values are selected for each Wikipedia page. Cosine similarity between the vector-based representations is computed and is used as the entities similarity score. Similarly, **OK**, a symmetric BM25 estimate that was specifically devised for estimating inter-document similarities [152], is used for comparing the entities' associated Wikipedia pages. The BM25 method was used for estimating the similarity between documents and query terms marked as entities [155, 158]. We are not familiar with a previous use of OK to estimate *inter-entity similarities* that are utilized for document retrieval.

The COS and OK measures defined above are query independent measures. Following past work [126] that demonstrated the merits of using query-dependent (sensitive) inter-document similarity measures we suggest to apply **QSSM1** and **QSSM3** [142] for estimating inter-entity similarities. The terms shared by the pages of two given entities are used for constructing a vector representation; the weight of each shared term is set to be the average of its weight in the two given entity pages. The cosine similarity between this vector and that of the query is than calculated. The result is used to scale the COS similarity measure in QSSM1 or is linearly interpolated with the COS score using a parameter γ in QSSM3. To the best of our knowledge, using these measures for estimating inter-entity similarities is novel to this study.

The Wikipedia Link-based Measure (**WLM**) utilizes the shared incoming and outgoing links within two entities corresponding Wikipedia pages to estimate their similarity [153]. Specifically, the Normalized Google Distance [34] is calculated

based on Wikipedia's links rather than on Google's search results. That distance is than transformed to similarity using a standard transformation. WLM can be computed efficiently and therefore it was used for the task of entity linking [57]. We are not familiar with methods utilizing it for the task of ad hoc document retrieval.

Finally, following past work [8, 99] we utilize the co-occurrence of an entity pair in different contexts to estimate its relatedness. **MI** is the first measure, utilizing the mutual information between the entities in the collection upon which search is performed, based on their appearances in common documents. In addition, we train a continuous bag-of-word (CBOW) model of Word2Vec[3] for creating entity embeddings. Specifically, we replace each entity annotation in a given text by the corresponding entity ID. The similarity value between two entities is set to the exponent of the cosine similarity between their vector representations. Following past work on terms [169], we train three models using three different document sets: the full Wikipedia repository (dump from 2014), top 5000 documents retrieved from Wikipedia or from the collection upon which search is performed using a standard term-based unigram language model retrieval. Each document set is used separately for training a model. The resulting similarity measures are named by the dataset used for learning the embeddings: Wikipedia Word2vec (**WW2V**), Local Wikipedia Word2vec (**LWW2V**) and Target collection Word2Vec (**TW2V**), respectively.

Evaluation measures and free parameters The two-tailed paired t-test with a 95% confidence level is used to determine statistically significant differences of the second-order cluster hypothesis test results.

The initial result list, D_{init}, defined in Section 6.1, is composed of 1000 documents. The list is retrieved using a standard term-based unigram language model retrieval [92]. The value of the Dirichlet smoothing parameter, μ, is set to 1000 in all retrieval methods. For relevance model construction $\mu = 0$ is also used. Following previous work [129], the number of entities composing S_r, the set of potentially relevant entities, and the number of documents used for relevance model construction are set to 100 and 50, respectively.

For determining the entity relevance, the ranking quality of various document lists is evaluated (see Section 6.2.1). We use Average precision at cutoff 1000 as an evaluation measure of the ranked lists. The values of the weighting parameters, λ_1 and λ_2, used for re-ranking the initial list D_{init} by three different retrieval methods are determined as follows. When retrieval methods E_Q and T_Q are used, λ_1 and λ_2 are set to a value in $\{0.0, 0.1, 0.2..., 1.0\}$. When $T_Q E_Q$ is used, we fix the relative weights of λ_1 and λ_2 by using 10-fold cross validation performed over all queries in the dataset. Query IDs are used to create the folds. The optimal parameter values for each of the 10 train sets are determined using a simple grid search applied to optimize MAP. The learned parameter values are then used for the queries in the

[3]https://code.google.com/archive/p/word2vec/

corresponding test fold. The value of the parameter λ_3, used for estimating the entity relevance, is determined by definition $\lambda_3 \stackrel{def}{=} 1 - (\lambda_1 + \lambda_2)$

The number of nearest neighbors, used for testing the hypothesis, k, is set to a value in $\{1, 2, 4, 9\}$. Some similarity measures are sparse (e.g., QSSM1) and therefore for some relevant entities there could be less than k neighbors as we do not consider neighbors with a 0 similarity value. In a case where only $l < k$ neighbors of a relevant entity are found, we randomly sample $k - l$ entities from the set S_r to complete the nearest neighbor set.

Given a configuration of λ_i values and the number of nearest neighbors, k, we optimize the second-order cluster hypothesis test result over all free parameters, i.e, the best test result, calculated by using all queries and relevant entities is presented.

Following previous work [49, 167, 169], for training all Word2Vec models we set the vector size to 300. The window size and the number of negative examples are set to a value in $\{8, 16\}$ and in $\{5, 10\}$, respectively. For calculating the QSSM3 measure we set the γ parameter value in $\{0.0, 0.1, ...1\}$. For the OK measure [152] we set b to 1.0 and k is set to a value in $\{4, 8, 12\}$.

6.3.2 Experimental results

Table 6.2 presents the second-order cluster hypothesis test results for each of the lists used for estimating the entity relevance, $D^{E_Q}_{base}$, $D^{T_Q}_{base}$ and $D^{T_Q E_Q}_{base}$, and for each of the evaluated similarity measures. The weight assigned to the evaluated entity-based score is set to $\lambda_3 = 0.1$ and the number of nearest neighbors, k is set to 4. We note that for queries with no entity markups, no relevance model is constructed by definition, and therefore the test result is set to zero. The number of queries without marked entities is 1, 3, 0 and 1 for ROBUST, WT10G, GOV2 and ClueBF, respectively.

We see that for vast majority of the proposed similarity measures and re-ranking methods, the hypothesis holds to a substantial extent, i.e., the average number of relevant entities among the k neighbors of a relevant entity is significantly higher than that obtained by randomly selecting entities from S_r excluding the examined entity. TW2V is the best performing similarity measure. This measure utilizes the cosine similarity between entity embeddings, learned by utilizing the text of documents retrieved in response to the query, from the collection upon which search is performed.

Generally, the similarity measures utilizing co-occurrence information: TW2V, WW2V, LWW2V, and MI yield the highest test results across all re-ranked lists and collections. When comparing the test results for the different measures utilizing Word2Vec-based entity embeddings we see that TW2V is the best performing measure while LWW2V is the worst. This finding echos results from document retrieval [49, 169] showing that utilizing locally trained embeddings, i.e., embedding learned by utilizing documents retrieved in response to the query, is more effective

Table 6.2: The second-order cluster hypothesis test results, $\lambda_3 = 0.1$ and $k = 4$. 'r' marks statistically significant difference with respect to *rand*.

	$D^{E_Q}_{base}$				$D^{T_Q}_{base}$				$D^{T_Q E_Q}_{base}$			
	ROBUST	WT10G	GOV2	ClueBF	ROBUST	WT10G	GOV2	ClueBF	ROBUST	WT10G	GOV2	ClueBF
rand	51.0	53.7	52.3	56.0	57.7	59.2	63.7	66.4	50.5	50.3	55.8	53.9
COS	59.9^r	64.3^r	62.3^r	64.3^r	65.7^r	70.3^r	71.8^r	75.2^r	59.1^r	63.8^r	64.9^r	65.0^r
QSSM1	53.3^r	57.0^r	55.3^r	57.3^r	58.8^r	62.2^r	63.6	66.9^r	50.7^r	52.8^r	56.5^r	54.9^r
QSSM3	60.0^r	64.4^r	62.3^r	64.3^r	65.7^r	70.4^r	71.8^r	75.3^r	59.1^r	63.8^r	64.9^r	65.0^r
OK	59.7^r	63.5^r	61.6^r	65.1^r	65.6^r	69.0^r	71.2^r	74.1^r	59.2^r	61.7^r	64.8^r	64.3^r
WLM	59.9^r	64.1^r	61.8^r	66.9^r	65.2^r	69.6^r	71.5^r	75.5^r	58.8^r	62.6^r	65.5^r	66.1^r
MI	**63.8^r**	65.1^r	62.7^r	65.8^r	67.9^r	**73.1^r**	74.7^r	**79.8^r**	**64.3^r**	65.2^r	66.8^r	**69.4^r**
WW2V	62.0^r	64.7^r	63.3^r	65.8^r	67.6^r	71.3^r	74.7^r	76.2^r	60.9^r	63.3^r	67.5^r	65.8^r
LWW2V	61.7^r	64.0^r	63.0^r	65.2^r	66.9^r	71.1^r	73.8^r	74.9^r	60.0^r	63.3^r	66.8^r	64.8^r
TW2V	63.3^r	**66.9^r**	**67.3^r**	**68.0^r**	**68.4^r**	71.4^r	**76.5^r**	76.1^r	61.6^r	**65.3^r**	**70.7^r**	66.4^r

for retrieval than utilizing globally trained embeddings, i.e., embeddings learned by utilizing some document collection. In addition, we see that entity embeddings learned by utilizing documents retrieved from the document collection upon which search is performed, are more effective than embeddings learned using documents in an external collection.

Similarity measures that utilize entity associated documents as well as entity associated links yield similar test results. OK is less effective than COS, QSSM3 and WLM. QSSM1 is the worst similarity measure in terms of the hypothesis test results. A possible reason for QSSM1's low performance is that QSSM1 is a sparse measure, i.e, the nearest neighbors of a relevant entity might be a mix of relevant entities and randomly selected entities.

Examining the test results for the different lists used for estimating the entity relevance we see that the best result is obtained when using the $D^{T_Q}_{base}$ list, i.e., term-only queries are used for re-ranking the initial list D_{init}. This result echos our previous findings (in Section 5.2.2.1), showing that using both entity and term-based information for retrieval is more effective than using either terms or entities. When entity-based queries are not used for creating the baseline ranking, it is likely that more entities would improve retrieval effectiveness.

Figure 6.1 presents the cluster hypothesis test results as a function of λ_3, the weight assigned to the evaluated entity-based document score. This parameter directly effects the number of relevant entities and therefore the test results. As expected, the test result as well as the number of relevant entities, decrease as λ_3 increases. The reason is that less entities are likely to improve retrieval effectiveness, when they are used for inducing a highly weighted query model than when a lower weight is used.

Figure 6.2 presents the cluster hypothesis test results as a function of k, the number of nearest neighbors examined for each relevant entity, given a fixed value of λ_3. The re-ranked list used for estimating the entity relevance is $D^{T_Q}_{base}$. As expected, the test result decrease as k increases. The reason is that selecting more

neighbors increases the probability of adding non-relevant entities. In most cases, the best performing measure is TW2V.

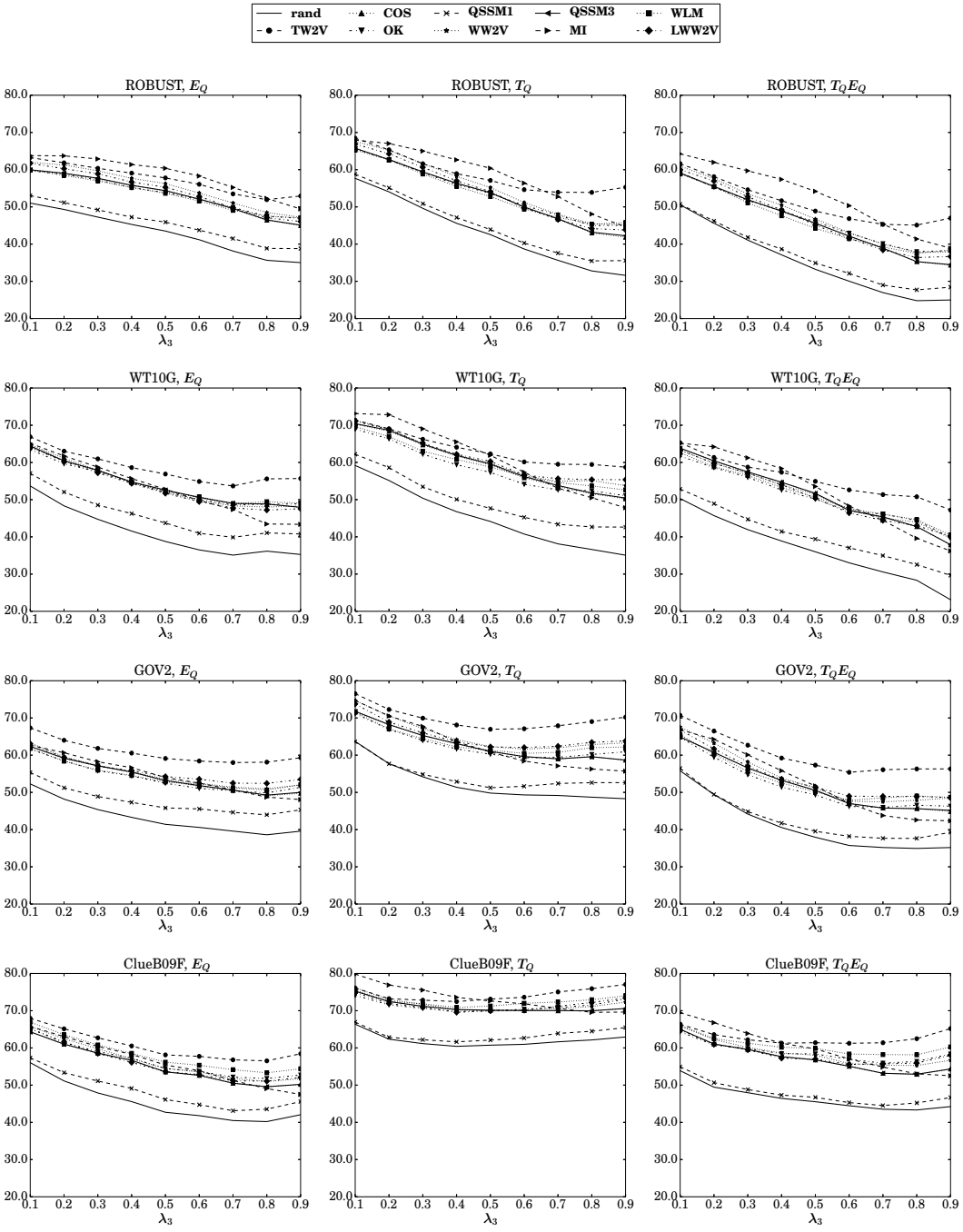

Figure 6.1: The effect of λ_3 on the second-order cluster hypothesis test, $k = 4$. The retrieval method used to re-rank D_{init} is denoted in each figure.

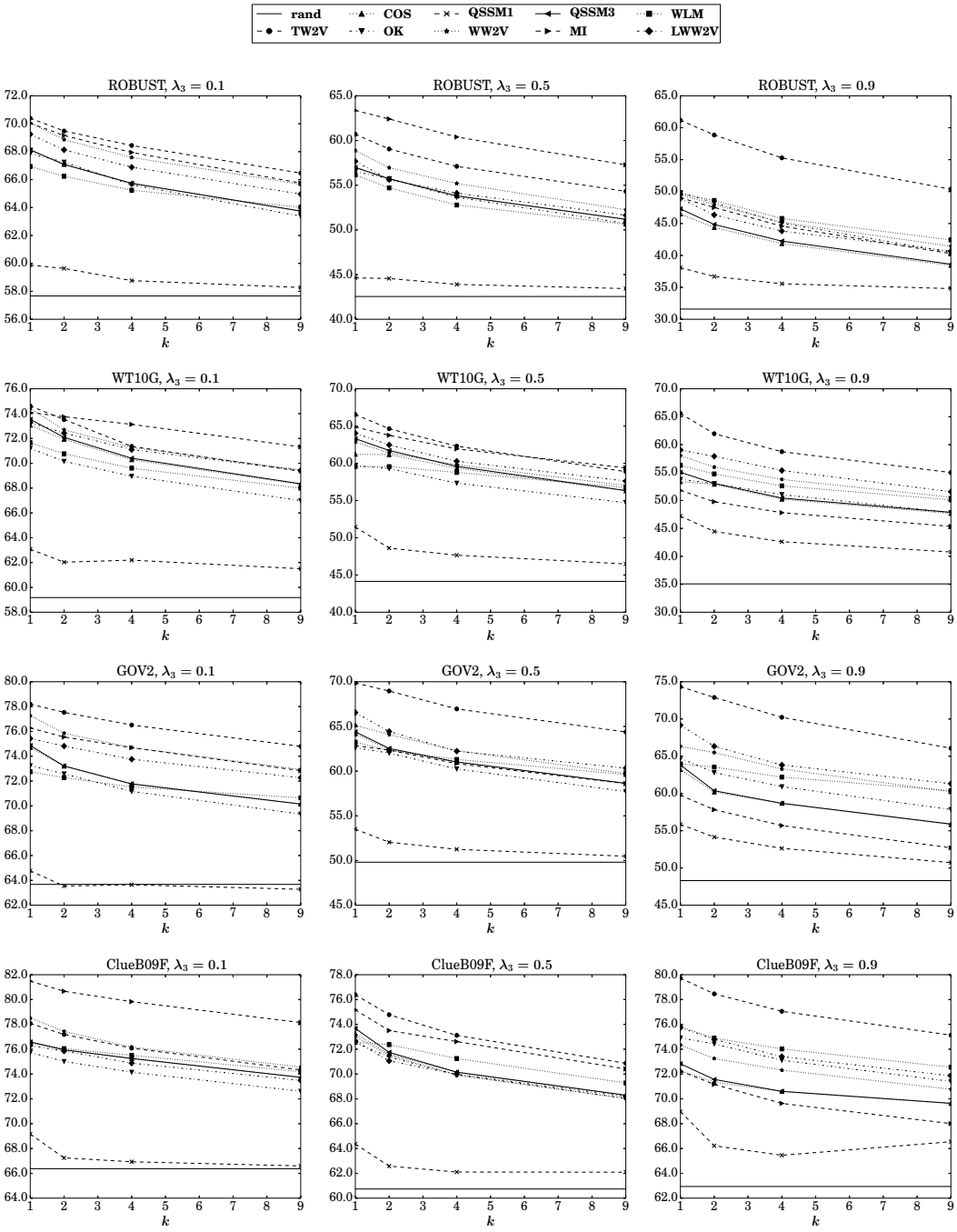

Figure 6.2: The effect of k on the second-order cluster hypothesis test. The re-ranked list used for estimating the entity relevance is D^{TQ}_{base}. Note: graphs are not to the same scale.

6.4 Inter-Entity Similarity-Based Query Models

Encouraged by findings of the second-order cluster hypothesis test we now turn to devise methods for inducing query models using inter-entity similarities. In the following we present three such methods.

6.4.1 The query similarity model

The basic assumption underlying our query similarity model is that entities *semantically similar* to the query are relevant to the underlying information need. We estimate the relevance for entities in some initial set, $\mathcal{S}$, by measuring their similarities with entities marked in the query. We use the estimates to rank the set; the highest ranked entities will be used for inducing a query model.

We use two initial entity sets. The first, denoted S_i, is the set of all entities in the entity repository which were marked at least once in a document in D_{init} with *any* confidence level[4]. Recall that D_{init} is a list of documents retrieved in response to the query by *some* retrieval method. The second set, S_r, is the c entities assigned the highest probability by the initial entity-only relevance model. (See Section 6.2.1 for details.). Generally, we denote an entity set that should be ranked with respect to the query by $\mathcal{S}$; e denotes an entity in $\mathcal{S}$ ($e \in \mathcal{S}$).

Let q and e' denote a query and an entity marked in the query with any confidence level, respectively ($e' \in q$). The similarity between e' and an entity e in the entity set ($e \in \mathcal{S}$), $sim(e', e)$[5], is estimated by *some* inter-entity similarity measure. The similarities are turned into "translation" probabilities $p_{sim}(e|e')$ as follows:

$$\hat{p}_{sim}(e|e') \stackrel{def}{=} \frac{sim(e, e')}{\sum_{e'' \in \mathcal{S}} sim(e'', e')}. \tag{6.4}$$

We now turn to define the query model estimation method utilizing these "translation" probabilities. The probability $p(e|\mathcal{M}^{\mathcal{S}}_{QS})$ of generating an entity $e \in \mathcal{S}$ from a query model $\mathcal{M}^{\mathcal{S}}_{QS}$ is estimated by:

$$\hat{p}(e|\mathcal{M}^{\mathcal{S}}_{QS}) \stackrel{def}{=} \sum_{e' \in q} \hat{p}_{sim}(e|e') \cdot p_{MLE}(e'|q); \tag{6.5}$$

$p_{MLE}(e'|q)$ is the maximum likelihood estimate (MLE) of e' with respect to q; $\hat{p}_{sim}(e|e')$ is a translation probability estimated as described above.

[4]We limit the number of entities used for inducing the query model due to computational considerations. Formally, the model can be induced using all entities marked in all documents in the corpus.

[5]Some similarity measures are sparse. Therefore we add an epsilon $\epsilon = 1e - 12$ to each similarity value provided by *some* similarity measure.

The query-model induction approach is referred to as Query Similarity ($\mathbf{QS}_{\mathcal{S}}$ in short) since inter-entity similarities are used to estimate entity generation probabilities. Depending on the entity set used, S_i or S_r, the method is denoted $\mathbf{QS_i}$ or $\mathbf{QS_r}$, respectively. Similarly, the induced query model is denoted $\mathcal{M}^i_{QS}$ or $\mathcal{M}^r_{QS}$, respectively.

6.4.2 The entity centrality query model

Our next query model induction methods is based on the notion of *entity centrality*. We assume that entities central to the initial set of entities which presumably pertain to the information need are more likely to be relevant than less central entities. We quantify centrality using inter-entity similarity measures. The initial set of entities is those assigned the highest probabilities by the relevance model: S_r.

We use $\mathcal{M}^r_{Cent}$ to denote a query model that is based on the centrality notion just described. Let R be the event of relevance. Our goal is to estimate $p(e|\mathcal{M}^r_{Cent},\mathrm{R})$ - the probability of generating the entity e from the query model $\mathcal{M}^r_{Cent}$ given that relevance is observed. We propose:

$$\hat{p}(e|\mathcal{M}^r_{Cent}, R) \stackrel{def}{=} \sum_{e' \in S_r} \hat{p}_{sim}(e|e', R) \cdot p(e'|R); \tag{6.6}$$

$p(e'|R)$ is entity's e' relevance probability estimated by utilizing the entity-only relevance model induced as described in Section 6.1.1. Specifically, we set: $p(e'|R) = \frac{p(e'|RM1)}{\sum_{e'' \in S_r} p(e''|RM1)}$; $p(e'|RM1)$ is the probability assigned to entity e' by the entity-only relevance model. $\hat{p}_{sim}(e|e', R)$ is a probability estimated by utilizing *some* inter-entity similarity measure as we detail below. We assume that the probability of e given e' is independent on R, i.e. $\hat{p}_{sim}(e|e', R) = \hat{p}_{sim}(e|e')$,

Inspired by work on inter-doc similarity [90] utilizing the nearest neighbors of a document to estimate its centrality in a document set, we use the nearest neighbors of an entity e' to estimate $\hat{p}_{sim}(e|e')$. Let $KNN(e')$ denote the set of k entities e which yield the highest inter-entity similarity, $sim(e, e')$, as determined by using some similarity measure. The similarities between entity e' and entities that are not in this set are set to zero. The probabilities are then estimated by sum-normalizing the similarity values:

$$\hat{p}_{sim}(e|e') \stackrel{def}{=} \begin{cases} \frac{sim(e,e')}{\sum_{e'' \in KNN(e')} sim(e'',e')} & \text{if } e \in KNN(e'); \\ 0 & \text{if } e \notin KNN(e'). \end{cases} \tag{6.7}$$

The proposed query-model estimation method is denoted Centrality ($\mathbf{Cent_r}$ in short) since the entity probability is estimated by considering its centrality in a set of potentially relevant entities. Specifically, entities that are included in the k nearest neighbors sets of many entities in S_r, are assigned with high query model generation probabilities using this method.

6.4.3 Query models induced from clusters of similar entities

The last inter-entity similarity-based query model we propose is inspired by the second-order cluster hypothesis, presented in Section 6.2. The hypothesis was that similar entities are relevant to the same requests.

We cluster the set S_r of c entities assigned the highest probabilities by the entity-only relevance model, using some inter-entity similarity measure. Specifically, an entity e in S_r, and its k nearest neighbors in S_r, are defined as a cluster. We use $Cl(S_r)$ to denote the set of clusters.

Herein, we use $\mathcal{M}^r_{Clust}$ to refer to the query model induced using the clusters in $Cl(S_r)$, R denotes the event of relevance. The probability assigned to entity e by the query model, given that relevance is observed, is defined as:

$$\hat{p}(e|\mathcal{M}^r_{Clust}, R) \stackrel{def}{=} \sum_{c \in Cl(S_r)} p(e|c, R) \cdot p(c|R); \tag{6.8}$$

We assume that $p(e|c, R) = p(e|c)$, i.e., the probability is independent on R, and set $p(e|c)$, the probability of generating entity e from a model induced from the cluster c, to be uniform[6]. $p(c|R)$ is the cluster's c relevance probability estimated by: $\hat{p}(c|R) \stackrel{def}{=} \max_{e \in c} p(e|\mathcal{M}^r_{QS})$; $p(e|\mathcal{M}^r_{QS})$ is the probability assigned to entity e by query similarity model, estimated as described in Section 6.4.1. Various entity-relevance features (e.g., entity centrality) as well as aggregation methods (e.g., geometric and average mean of the relevance estimates for the entities in the cluster) were utilized in various experiments. The estimate $\hat{p}(c|R) \stackrel{def}{=} \max_{e \in c} p(e|\mathcal{M}^r_{QS})$ was found to be most effective for retrieval and therefore it is presented in this work. The proposed cluster-based query model estimation method is denoted Cluster ($\mathbf{Clust_r}$ in short). We note that when $k = 0$, i.e., only singleton clusters containing a single entity are used, the $Clust_r$ method reduces to QS_r.

6.4.4 Integration with the relevance model

Let $\mathcal{M}_P$ denote one of the query models in $\{\mathcal{M}^i_{QS}, \mathcal{M}^r_{QS}, \mathcal{M}^r_{Cent}, \mathcal{M}^r_{Clust}\}$ which were described in Sections 6.4.1, 6.4.2 and 6.4.3. P denotes the method used for inducing the query model $\mathcal{M}_P$. We next integrate these query models with the relevance model.

We follow common practice in work on pseudo-feedback-based query models defined over terms [91], and clip both the relevance model and our suggested models. The probabilities of all but the ν entities assigned the highest probability by a given query model are set to zero and the remaining probabilities are sum

[6]We experimented with additional estimation methods, for example utilizing the cluster entities relevance model probabilities. These results are omitted since no improvements in retrieval effectiveness were observed.

normalized to create $\hat{p}_{clip}(e|\mathcal{M}_P)$ and $\hat{p}_{clip}(e|RM1)$; ν is a free parameter. The models are then integrated using a parameter α:

$$p(e|\mathcal{M}_P, RM1) \stackrel{def}{=} \alpha \hat{p}_{clip}(e|\mathcal{M}_P) + (1-\alpha)\hat{p}_{clip}(e|RM1). \quad (6.9)$$

Then, $p(e|\mathcal{M}_P, RM1)$ is clipped by setting to zero the probabilities of all entities except for the m assigned the highest probability (yielding $p_{clip}(e|\mathcal{M}_P, RM1)$); m is a free parameter. We denote the $RM1$ induction method by **RM**. Then, the integrated query model is denoted $\mathcal{M}_{\mathbf{RM-P}}$ and the proposed query model estimation method is denoted $\mathbf{RM-P}$, as we integrate the model $\mathcal{M}_P$ induced by method P, with a relevance model $RM1$ induced by method RM. The resulting query model induction methods are: $\mathbf{RM-QS_i}$, $\mathbf{RM-QS_r}$, $\mathbf{RM-Cent_r}$ and $\mathbf{RM-Clust_r}$.

6.4.5 Term-based query models estimation

In this work we focus on evaluating the merits of utilizing inter-*entity* similarities to induce query models. There has been much work on utilizing inter-term similarities for document retrieval (e.g., [91, 49, 167]). Evaluating the effectiveness of utilizing both term similarities and entity similarities to induce query models is an interesting research direction that we leave for future work.

In the following, we only use common estimation methods for inducing the term-based query models. Specifically, we use the maximum likelihood estimate of the term-only query STLM model, $p_{MLE}(w|q) \stackrel{def}{=} \frac{c(w)}{l(q)}$; w is a term, $c(w)$ is the count of term w in the query q, $l(q)$ is the term-only query length. We also use the term-only relevance model, $p(w|RM1)$, induced as described in Section 6.1.1. The methods used for inducing these two query models are denoted **Q** and **RM**, respectively. They also serve for inducing equivalent entity-only query models.

6.4.6 Query anchoring

A query model induced by some query model estimation method can drift away from the information need [116]. To mitigate that risk, we use a common *query anchoring* technique [133, 1, 171] which uses the original query as an anchor when utilizing the proposed query model. Specifically, the query model is interpolated with a model of the original query to created a new query model.

Formally, let P denote a query model estimation method used to induce some query model $\mathcal{M}_P$, t denotes a token which can be either an entity (e) or a term (w). We use an unsmoothed maximum likelihood estimate of the entity-only or term-only query STLM model, denoted $p_{MLE}(t|q)$. The new query model is an interpolation of the two models:

$$p(t|q, \mathcal{M}_P) \stackrel{def}{=} (1-\beta)p_{MLE}(t|q) + \beta p(t|\mathcal{M}_P); \quad (6.10)$$

β is a free parameter which can depend on the token type, i.e., the values of β when t is an entity or a term can be different.

We use the original notation P, to denote a query model estimation method which utilizes query anchoring, since query anchoring is applied for all our suggested query models. As we detail below, the resulting entity-based and term-based query model probability distributions, $p(e|q, \mathcal{M}_P)$ and $p(w|q, \mathcal{M}_P)$ are used as estimates of θ_q^{ent} and θ_q^{term} in Equation 6.1, respectively.

6.4.7 Terms and entities integration

To assign documents with retrieval scores, combinations of term-based and entity-based query models, induced by methods described above are used. As explained in Section 6.1, given the term-only and entity-only query models, θ_q^{term} and θ_q^{ent}, and the term-only and entity-only document models, θ_d^{term} and θ_d^{ent}, the document score is computed by:

$$S(d;q) \stackrel{def}{=} \lambda CE\left(\theta_q^{term} \mid\mid \theta_d^{term}\right) + (1-\lambda) CE\left(\theta_q^{ent} \mid\mid \theta_d^{ent}\right); \tag{6.11}$$

$CE(\cdot \mid\mid \cdot)$ is the cross-entropy measure, λ is a free parameter.

To formally define a retrieval method that corresponds to the integration of retrieval scores assigned by using different entity and term query models, we use the notation $T_{pw}E_{pe}$; T and E denote terms and entities, respectively, pe is an entity-only query model estimation method in the set $P_e \stackrel{def}{=} \{Q, RM, QS_i, QS_r, Cent_r, Clust_r, RM-QS_i, RM-QS_r, RM-Cent_r, RM-Clust_r\}$, pw is a term-only query model estimation method in the set $P_w \stackrel{def}{=} \{Q, RM\}$. In some cases only a term query model or an entity query model is used. The retrieval method is then denoted T_{pw} or E_{pe}, respectively.

The resulting retrieval methods are: $\mathbf{E_Q}$, $\mathbf{T_Q}$, $\mathbf{E_{RM}}$, $\mathbf{T_QE_Q}$, $\mathbf{T_QE_{RM}}$, $\mathbf{T_QE_{QS_i}}(\cdot)$, $\mathbf{T_QE_{QS_r}}(\cdot)$, $\mathbf{T_QE_{Cent_r}}(\cdot)$, $\mathbf{T_QE_{Clust_r}}(\cdot)$, $\mathbf{T_QE_{RM-QS_i}}(\cdot)$, $\mathbf{T_QE_{RM-Cent_r}}(\cdot)$, $\mathbf{T_QE_{RM-QS_r}}(\cdot)$, $\mathbf{T_QE_{RM-Clust_r}}(\cdot)$. The specific similarity measure used for estimating inter-entity similarities is denoted inside the parentheses.

The retrieval methods utilizing the term-based relevance model are: $\mathbf{T_{RM}}$, $\mathbf{T_{RM}E_Q}$, $\mathbf{T_{RM}E_{RM}}$. Methods using the term-based relevance model together with inter-entity similarity based query models did not yield improved retrieval performance when compared with the methods presented below and therefore are omitted from the evaluation.

6.5 Evaluation of the Inter-Entity Similarity-Based Query Models

We now turn to evaluate the effectiveness of our proposed methods for inducing inter-entity similarity-based query models.

6.5.1 Experimental setup

Baselines We use standard term-based unigram language model retrieval [92] for retrieving the initial result list of 1000 documents, denoted D_{init}. This list is re-ranked using the retrieval methods suggested in Section 6.4.7. The standard term-based unigram language model retrieval method is denoted by T_Q, following our methods naming scheme.

The first group of baselines we compare with utilize the maximum likelihood estimate of the query STLM language model for inducting the term and entity query models. These are the methods T_Q, E_Q and $T_Q E_Q$, referred to as TermsLM, STOEnt and F-ST in Chapter 5, respectively. T_Q is the standard term-based unigram language model retrieval [92]. The entity-based models, utilizing entity-only information (E_Q) or integration of entity-only and term-only information ($T_Q E_Q$) were described in Chapter 5. We note that the methods E_Q and $T_Q E_Q$ are used here to re-rank an initially retrieved list while in their original report [129], they were used to rank the entire corpus. Also, wider range of smoothing parameter values was used for evaluating these retrieval methods in the original report [129]. As detailed below, in this chapter we fix the smoothing parameter to a value which yields effective retrieval across all collections.

An additional set of baselines is composed of methods utilizing the entity-only, term-only or entity+term relevance models [93]. These are the E_{RM}, T_{RM}, $T_Q E_{RM}$, $T_{RM} E_Q$ and $T_{RM} E_{RM}$ methods. Methods that are similar in spirit were proposed and evaluated in Section 5.2.3, however there is one important difference. In Section 5.2.3, STLM language models, which integrate term-based and entity-based information at the *language model level*, were utilized for inducing a relevance model. The methods suggested in this chapter integrate term and entity relevance models at the *retrieval score level* (See Equation 6.1 on page 53). As will be shown below, integration at the retrieval-score level yields much better performance than integration at the language-model level. The methods most similar in spirit to the $T_Q E_{RM}$, $T_{RM} E_Q$ and $T_{RM} E_{RM}$ methods proposed in this section were referred to as RMST and RMST-ST in Chapter 5.

Retrieval methods utilizing integration of scores attained by using multiple term-only and entity-only expanded query forms were recently proposed by Dalton et al. [42]. These models are similar in spirit to $T_{RM} E_{RM}$ which serve as a baseline in our work. Additional group of previously proposed retrieval methods utilize Wikipedia pages of entities associated with the query for inducing a *term*-based query model [156, 160]. We show in Section 5.2.3 that integrating entity-based an term-based relevance models yields better retrieval performance than utilizing term-based query models induced from Wikipedia pages. This result echos findings from previous work [42]. Also, as noted, the relevance model based methods we use ($T_Q E_{RM}$, $T_{RM} E_Q$ and $T_{RM} E_{RM}$) are more effective than relevance model-based methods evaluated in Section 5.2.3, due to integration at the score level instead of at the language model level. We therefore only compare with methods utilizing the entity, term or entity+term relevance models as suggested above.

Evaluation measures and free parameters Mean average precision at cutoff 1000 (MAP), precision of the top 10 documents (p@10) and NDCG@10 (NDCG) serve as evaluation measures. Statistically significant performance differences are determined using the two-tailed paired t-test with a 95% confidence level.

The free parameter values of *all* retrieval methods are set using 10-fold cross validation performed over the queries in a dataset. Query IDs are used to create the folds. The optimal parameter values for each of the 10 train sets are determined using a simple grid search applied to optimize MAP. The learned parameter values are then used for the queries in the corresponding test fold.

The value of the Dirichlet smoothing parameter, μ, is set to 1000 in all retrieval methods. For relevance model construction $\mu = 0$ is also used. Following past work [129], the number of entities in S_r, the set of entities assigned the highest probabilities by the entity-only relevance model, and the number of documents used for relevance model construction are set to 100 and 50, respectively. The values of λ, α and β, used for integrating different types of query models (in Equations 6.1, 6.10 and 6.9), are selected from $\{0.0, 0.1, 0.2..., 1.0\}$.

The number of tokens ν used for clipping all query models is set to a value in $\{10, 25, 50, 100\}$. When both entity and term relevance models are used the number of clipped tokens is coupled to the same value. When an entity-based relevance model is integrated with a query model induced by one of the methods: QS_i, QS_r or $Cent_r$, the models are clipped using the same value in $\{10, 25, 50, 100\}$.

Following previous work on utilizing clusters of similar entities for entity retrieval [128] the cluster size k, used by the $Clust_r$ method, is set to 5 and the number of nearest neighbors selected for each entity in S_r is set to 4. To induce a query model using the $Clust_r$ method we select the number of clusters assigned the highest cluster probability ($\hat{p}(c|R)$) from $\{1, 5, 10, 20\}$. When interpolating the query model induced by the $Clust_r$ method with the entity-only relevance model, the number of clusters utilized is coupled with the number of tokens used for clipping as follows: $\{1-10, 5-25, 10-50, 20-100\}$; the first number in each pair denotes the number of clusters used for inducing an entity-based query model, and the second denotes the number of tokens used for clipping the entity-based relevance model.

6.5.2 Experimental results

Table 6.3 presents a comparison between methods utilizing the term-only query (T_Q), the entity-only query (E_Q), the entity-only relevance model (E_{RM} and $T_Q E_{RM}$) or the entity-only query model induced by applying the QS_r method ($T_Q E_{QS_r}(\cdot)$). We compare the retrieval performance of methods using different inter-entity similarity measures to identify measures that are most effective for retrieval. These measures will be used for additional comparisons between query model induction methods proposed in this chapter. The query model induction method, QS_r, was selected since it is the most effective for retrieval, as will be shown below.

Consistent with findings in Chapter 5, our first observation based on Table 6.3

is that using a combination of term-only and entity-only information at the score level ($T_Q E_Q$ which is equivalent to F-ST) significantly improves retrieval performance with respect to using entity-only or term-only information alone (E_Q[7] or T_Q, equivalent to STOEnt or TermsLM, respectively). It is interesting to note that employing E_Q results in much better retrieval effectiveness than that presented in Section 5.2.2.1 for STOEnt. The reason is that in this setup, E_Q is used for reranking an initially retrieved result list, while in Chapter 5 STOEnt was used for retrieving documents from the entire corpus. An additional interesting observation, consistent with findings from Chapter 5, is that methods utilizing entity-only relevance models are more effective than methods using the query alone (Compare E_Q with E_{RM} and $T_Q E_Q$ with $T_Q E_{RM}$.)

We see in Table 6.3 that methods utilizing inter-entity similarity-based query models yield improved retrieval performance with respect to the two highly effective baselines: $T_Q E_Q$ and $T_Q E_{RM}$. The analysis is performed by separately comparing each method utilizing some inter-entity similarity measure with each of the proposed baselines; i.e., the number of cases in which a method $T_Q E_{QS_r}(\cdot)$, utilizing some inter-entity similarity measure, is more effective than a given baseline is counted. We see that for all similarity measures, the differences between $T_Q E_{QS_r}(\cdot)$ and $T_Q E_Q$ are statistically significant in 30 percent or more of the comparisons. The differences between the $T_Q E_{QS_r}(\cdot)$ methods and $T_Q E_{RM}$ are less significant, as expected. $T_Q E_{QS_r}(\cdot)$ utilizing the QSSM1 measure is the most effective method when compared with $T_Q E_{RM}$, with statistically significant differences in 40 percent of the comparisons. The differences between methods utilizing the TW2V, LWW2V and COS similarity measures and $T_Q E_{RM}$ are statistically significant in 25, 16 and 16 percent of the comparisons, respectively. Methods utilizing other inter-entity similarity measures perform worse or not significantly better than the $T_Q E_{RM}$ baseline.

To rank inter-entity similarity measures we compare a method $T_Q E_{QS_r}(\cdot)$, utilizing a given measure, with the methods $T_Q E_{QS_r}(\cdot)$, utilizing all other measures. The most effective inter-entity similarity measure is QSSM1, as it outperforms other measures in most relevant comparisons. It is interesting to note that QSSM1 was found to be the least effective measure by the second-order cluster hypothesis test in Section 6.3. The finding in Section 6.3 is explained by the sparseness of the QSSM1 measure. For evaluating the cluster hypothesis test, nearest neighbors of relevant entities were selected, based on similarity estimates provided by the QSSM1 measure. As noted in Section 6.3, some of these clusters were created randomly and this noise affected the results. The use of QSSM1 for measuring inter-entity similarities is novel to this study.

The second best performing similarity measure is COS. This measure compares Wikipedia pages of two given entities and was previously used for estimating inter-entity similarities that were utilized for document retrieval [103]. It is interesting

[7]For queries with no marked entities, all methods rely on the maximum likelihood estimate of the term-only query.

to note that QSSM1, which is a query sensitive similarity measure, outperforms COS. QSSM3, which considers query sensitivity differently than QSSM1 is less effective than both these measures, even though the differences between methods utilizing these three measures are not statistically significant.

Consistently with findings from Section 6.3.2 regarding the second-order cluster hypothesis test, utilizing entity embeddings for estimating inter-entity similarities is effective for retrieval, as the group of Word2Vec-based measures is next in the measure rank. Embeddings learned from query dependent collections (i.e., documents retrieved in response to a query) are more effective for retrieval than embeddings learned from a general collection (i.e., Wikipedia). Specifically, methods utilizing inter-entity similarities estimated by the TW2V and LWW2V measures are more effective than methods utilizing inter-entity similarities estimated by WW2V.

The retrieval effectiveness of methods utilizing the remaining inter-entity similarity measures, OK, MI and WLM is similar. WLM, which utilizes entity links for estimating similarities, is the worst performing measure.

Based on the findings in Table 6.3 discussed above, we select the COS and QSSM1 measures for additional comparisons of our proposed retrieval methods. We note that these similarity measures were also found to be highly effective when utilized by additional entity-based query models estimation methods. We omit the full results to avoid cluttering the discussion.

Table 6.3: Comparing various inter-entity similarity measures utilized by the QS_r method. e, t, r, s and $*$ mark significant difference with E_Q, T_Q, E_{RM}, T_QE_Q and T_QE_{RM} respectively.

	ROBUST			WT10G			GOV2			ClueBF		
	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG
E_Q	21.7	37.6	38.4	18.5	27.9	27.5	26.3	49.6	39.0	16.8	33.6	25.1
T_Q	25.2^{e}	42.7^{e}	43.9^{e}	19.0	28.0	30.9	29.5^{e}	53.1	44.9^{e}	17.9	32.1	22.9
E_{RM}	24.2^{e}	38.6^{t}	38.9^{t}	19.1	28.0	26.6	27.2^{e}	49.7	38.1^{t}	18.0^{e}	35.6	27.8^{e}
T_QE_Q	$27.9^{e,t}_{r}$	$46.7^{e,t}_{r}$	$\mathbf{48.2^{e,t}_{r}}$	$22.4^{e,t}_{r}$	$31.7^{e,t}_{r}$	34.2^{e}_{r}	$32.4^{e,t}_{r}$	$58.7^{e,t}_{r}$	$48.9^{e,t}_{r}$	$21.3^{e,t}_{r}$	$38.6^{e,t}_{r}$	28.5^{t}_{r}
T_QE_{RM}	$28.7^{e,t}_{r,s}$	$46.8^{e,t}_{r}$	$47.6^{e,t}_{r}$	$21.5^{e,t}_{r}$	$31.8^{e,t}_{r}$	33.5^{e}_{r}	$32.9^{e,t}_{r}$	$59.5^{e,t}_{r}$	$49.0^{e,t}_{r}$	$22.9^{e,t}_{r,s}$	$43.6^{e,t}_{r,s}$	$33.4^{e,t}_{r,s}$
$T_QE_{QS_r}(COS)$	$28.6^{e,t}_{r,s}$	$\mathbf{47.3^{e,t}_{r}}$	$\mathbf{48.2^{e,t}_{r}}$	$22.1^{e,t}_{r}$	$33.9^{e,t}_{r}$	34.6^{e}_{r}	$\mathbf{33.5^{e,t}_{r,s}}$	$\mathbf{61.1^{e,t}_{r}}$	$\mathbf{50.9^{e,t}_{r}}$	$23.9^{e,t}_{r,s,*}$	$45.7^{e,t}_{r,s}$	$35.3^{e,t}_{r,s,*}$
$T_QE_{QS_r}(QSSM1)$	$\mathbf{28.8^{e,t}_{r,s}}$	$\mathbf{47.3^{e,t}_{r}}$	$\mathbf{48.2^{e,t}_{r}}$	$\mathbf{22.7^{e,t}_{r,*}}$	$\mathbf{34.4^{e,t}_{r,s,*}}$	$\mathbf{35.2^{e,t}_{r}}$	$33.1^{e,t}_{r}$	$59.7^{e,t}_{r}$	$49.5^{e,t}_{r}$	$\mathbf{24.2^{e,t}_{r,s,*}}$	$\mathbf{46.2^{e,t}_{r,s,*}}$	$\mathbf{35.7^{e,t}_{r,s,*}}$
$T_QE_{QS_r}(QSSM3)$	$28.6^{e,t}_{r,s}$	$46.9^{e,t}_{r}$	$48.1^{e,t}_{r}$	$22.3^{e,t}_{r}$	$32.9^{e,t}_{r}$	34.5^{e}_{r}	$33.4^{e,t}_{r,s}$	$60.9^{e,t}_{r}$	$50.5^{e,t}_{r}$	$24.0^{e,t}_{r,s,*}$	$45.9^{e,t}_{r,s}$	$35.1^{e,t}_{r,s}$
$T_QE_{QS_r}(OK)$	$28.4^{e,t}_{r}$	$46.7^{e,t}_{r}$	$47.4^{e,t}_{r}$	$21.6^{e,t}_{r}$	$32.1^{e,t}_{r}$	32.4^{e}_{r}	$32.4^{e,t}_{r,*}$	$59.7^{e,t}_{r}$	$49.8^{e,t}_{r}$	$23.8^{e,t}_{r,s,*}$	$45.7^{e,t}_{r,s}$	$35.2^{e,t}_{r,s}$
$T_QE_{QS_r}(WLM)$	$28.6^{e,t}_{r,s}$	$47.0^{e,t}_{r}$	$48.0^{e,t}_{r}$	21.3^{e}	$31.3^{e,t}$	32.9^{e}_{r}	$32.4^{e,t}_{r,*}$	$58.3^{e,t}_{r}$	47.9^{e}_{r}	$22.7^{e,t}_{r,s}$	$43.2^{e,t}_{r,s}$	$33.3^{e,t}_{r,s}$
$T_QE_{QS_r}(MI)$	$28.6^{e,t}_{r,s}$	$47.1^{e,t}_{r}$	$48.1^{e,t}_{r}$	$21.6^{e,t}_{s}$	31.0^{t}	33.3^{e}_{r}	$32.5^{e,t}_{r}$	$59.1^{e,t}_{r}$	$49.6^{e,t}_{r}$	$23.2^{e,t}_{r,s}$	$44.1^{e,t}_{r,s}$	$33.7^{e,t}_{r,s}$
$T_QE_{QS_r}(WW2V)$	$28.6^{e,t}_{r,s}$	$47.0^{e,t}_{r}$	$47.6^{e,t}_{r}$	$21.4^{e,t}_{r}$	$31.9^{e,t}_{r}$	32.7^{e}_{r}	$32.3^{e,t}_{r,*}$	$58.9^{e,t}_{r}$	$48.6^{e,t}_{r}$	$22.5^{e,t}_{r,s}$	$42.9^{e,t}_{r,s}$	$33.2^{e,t}_{r,s}$
$T_QE_{QS_r}(LWW2V)$	$28.3^{e,t}_{r}$	$47.1^{e,t}_{r}$	$47.5^{e,t}_{r}$	$22.3^{e,t}_{r,*}$	$34.0^{e,t}_{r,s,*}$	34.4^{e}_{r}	$32.9^{e,t}_{r}$	$59.9^{e,t}_{r}$	$49.6^{e,t}_{r}$	$23.2^{e,t}_{r,s}$	$44.9^{e,t}_{r,s}$	$34.4^{e,t}_{r,s}$
$T_QE_{QS_r}(TW2V)$	$28.7^{e,t}_{r,s}$	$47.0^{e,t}_{r}$	$47.9^{e,t}_{r}$	$22.5^{e,t}_{r,*}$	$32.7^{e,t}_{r}$	$34.7^{e,t}_{r,*}$	$32.9^{e,t}_{r}$	$59.8^{e,t}_{r}$	$49.7^{e,t}_{r}$	$23.6^{e,t}_{r,s}$	$45.1^{e,t}_{r,s}$	$35.1^{e,t}_{r,s,*}$

We now turn to compare and evaluate different methods for inducing entity-based query models, proposed in Section 6.4: QS_i, QS_r, $Cent_r$ and $Clust_r$. The performance numbers of each of the methods utilizing query models induced by these methods are compared with those of our proposed baselines, T_QE_Q and

T_QE_{RM}. Specifically, we count the number of statistically significant differences between each of the baselines and each of the proposed methods, for each of the similarity measures: COS and QSSM1.

In Table 6.4 we see that in most relevant comparisons, both $T_QE_{QS_r}(\cdot)$ and $T_QE_{Clust_r}(\cdot)$ methods, utilizing both similarity measures, COS and QSSM1, outperform the T_QE_Q baseline. The differences are statistically significant in between 40 to 50 percent of the comparisons. The differences between these methods and T_QE_{RM} are smaller. Our proposed methods are statistically significantly better than T_QE_{RM} in between 8 to 40 percent of the comparisons.

Recall that the QS_r query model induction method is a special case of the $Clust_r$ method. Since most of the statistically significant differences between the $T_QE_{QS_r}(\cdot)$ and $T_QE_{Clust_r}(\cdot)$ methods are in favor of $T_QE_{QS_r}(\cdot)$, we point out that given the specific cluster relevance estimation method proposed in Section 6.4.3, the dominant factor affecting retrieval performance is the query similarity estimate and not the entity clusters used for inducing the query model.

The methods $T_QE_{Cent_r}(\cdot)$ and $T_QE_{QS_i}(\cdot)$, utilizing query models induced by $Cent_r$ and QS_i, respectively, outperform T_QE_Q in the majority of comparisons, for both similarity measures. For both methods and measures, between 25 to 33 of the differences are statistically significant. Both methods are not statistically significantly better with respect to T_QE_{RM}, when utilizing either of the examined inter-entity similarity measures.

An additional interesting observation from Table 6.4, is that $T_QE_{QS_r}()$ outperforms $T_QE_{QS_i}()$. QS_r ranks entities in S_r, the set of c entities assigned the highest probabilities by the entity-only relevance model. QS_i ranks entities in a set of all entities which were marked at least once in a document in D_{init} with *any* confidence level. Both QS_r and QS_i use the same ranking method. In most comparisons, retrieval methods utilizing query models induced by QS_r yields better retrieval effectiveness than retrieval methods utilizing query models induced by QS_i. The differences are statistically significant in almost half of the comparisons, for both similarity measures. We explain performance differences by the quality of the entity set being ranked. Presumably, the percentage of entities in S_r, which are relevant with respect to the query, is higher than the percentage of relevant entities in a set composed of all entities marked in D_{init}.

The integration of the similarity-based query models with entity-only relevance model is only effective in a few cases, most of them are for the ClueBF dataset. We note that QS_r, $Cent_r$ and $Clust_r$ methods are essentially methods for re-ranking the entity-only relevance model. Re-ranking the relevance model can be viewed as an alternative approach to the models interpolation approach presented in Section 6.4.4 and evaluated in Table 6.4 (methods $T_QE_{RM-QS_r}(\cdot)$, $T_QE_{RM-Cent_r}(\cdot)$ and $T_QE_{RM-Clust_r}(\cdot)$). The former approach turns out to be more effective for retrieval.

Comparison with the term-only relevance model So far we compared methods utilizing various types of entity-based query models. Table 6.5 shows a

Table 6.4: Comparing entity-based query model induction methods. 's' and '$*$' mark significant difference with $T_Q E_Q$ and $T_Q E_{RM}$, respectively. 'l' marks significant difference with $T_Q E_{QS_r}(\cdot)$ when using the same similarity measure.

	ROBUST			WT10G			GOV2			ClueBF		
	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG
$T_Q E_Q$	27.9	46.7	48.2	22.4	31.7	34.2	32.4	58.7	48.9	21.3	38.6	28.5
$T_Q E_{RM}$	28.7_s	46.8	47.6	21.5	31.8	33.5	32.9	59.5	49.0	22.9_s	43.6_s	33.4_s
$T_Q E_{QS_i}(COS)$	28.2_s	47.1	48.4	22.3	32.3	34.0	32.5^l	58.6^l	48.6^l	22.9^l_s	44.1_s	33.6^l_s
$T_Q E_{Cent_r}(COS)$	28.4_s	47.1	48.0	21.4	32.2	33.6	32.7^l	59.3	49.4	23.5_s	43.7^l_s	33.9^l_s
$T_Q E_{QS_r}(COS)$	28.6_s	47.3	48.2	22.1	33.9	34.6	$\mathbf{33.5_s}$	61.1	50.9	$23.9_{s,*}$	45.7_s	$35.3_{s,*}$
$T_Q E_{Clust_r}(COS)$	28.4_s	**47.6**	$\mathbf{48.8}_*$	$\mathbf{23.8^l_{s,*}}$	$\mathbf{34.5_{s,*}}$	**35.9**	32.7^l	60.4	50.3	23.7_s	45.0_s	34.3_s
$T_Q E_{RM-QS_i}(COS)$	28.8_s	46.9	47.9	22.1	32.9	34.2	32.8^l	58.9	47.9^l	22.7^l_s	42.7^l_s	33.5^l_s
$T_Q E_{RM-Cent_r}(COS)$	$28.8_{s,*}$	46.9	47.6	21.5	31.9	33.2	32.9	59.6	49.0	23.3^l_s	43.5^l_s	33.5^l_s
$T_Q E_{RM-QS_r}(COS)$	28.8_s	47.3	48.2	22.4_*	33.4_s	35.1	33.3_s	60.5	50.2	$23.9_{s,*}$	45.3_s	35.0_s
$T_Q E_{RM-Clust_r}(COS)$	28.4	46.4	47.4	$23.7_{s,*}$	$\mathbf{34.5_{s,*}}$	$\mathbf{35.9}_*$	$\mathbf{33.5_{s,*}}$	$\mathbf{61.3_s}$	$\mathbf{51.1_{s,*}}$	23.5_s	44.3_s	34.1^l_s
$T_Q E_{QS_i}(QSSM1)$	28.3^l	47.1	48.5	22.7	33.0	34.8	32.3^l	57.9	47.5^l	22.6^l_s	42.1^l_s	32.7^l_s
$T_Q E_{Cent_r}(QSSM1)$	28.0^l_*	46.5	47.5_s	22.0	32.5	33.5	32.9	59.3	49.4	22.9^l_s	44.5_s	34.2_s
$T_Q E_{QS_r}(QSSM1)$	28.8_s	47.3	48.2	22.7_*	$34.4_{s,*}$	35.2	33.1	59.7	49.5	$\mathbf{24.2_{s,*}}$	$\mathbf{46.2_{s,*}}$	$\mathbf{35.7_{s,*}}$
$T_Q E_{Clust_r}(QSSM1)$	28.3^l_s	46.7	47.7	22.5_*	33.1	34.5	33.0_s	60.1	50.0	23.1^l_s	43.7^l_s	32.9^l_s
$T_Q E_{RM-QS_i}(QSSM1)$	28.7_s	47.3	48.1	22.4	32.7	34.1	33.0	58.8	48.7	23.0^l_s	44.4_s	34.2_s
$T_Q E_{RM-Cent_r}(QSSM1)$	28.7_s	46.6	47.5	21.9	32.8	33.4	32.9	59.3	48.9	23.4^l_s	44.3_s	33.8^l_s
$T_Q E_{RM-QS_r}(QSSM1)$	$\mathbf{29.0_s}$	47.2	48.1	22.6_*	34.3_*	35.0	33.1	59.8	49.3	$24.1_{s,*}$	45.8_s	35.2_s
$T_Q E_{RM-Clust_r}(QSSM1)$	28.7_s	47.1	47.6	22.3	33.8	35.0	32.9	58.9	48.9	$23.8_{s,*}$	44.8_s	34.0_s

comparison of the best performing inter-entity similarity-based methods, according to the analysis presented above, $T_Q E_{QS_r}(\cdot)$ and $T_Q E_{Clust_r}(\cdot)$, with methods utilizing the term-based relevance model.

We observe that methods utilizing the term-based relevance model on top of the entity and term-based queries are more effective than methods utilizing the entity-based relevance model (compare T_{RM} with E_{RM} and $T_{RM} E_Q$ with $T_Q E_{RM}$). For two collections, integrating the term-based and entity-based relevance model results in improvements in retrieval performance with respect to using each model alone (compare $T_{RM} E_{RM}$ with $T_Q E_{RM}$ and $T_{RM} E_Q$). It turns out that integrating term-based and entity-based information at the *retrieval score* level yields improved retrieval results compared with integration at the *language model* level. We refer the reader back to Table 5.8 on page 50. The performance numbers obtained when using the RMST-ST method are lower, for some collections, than those obtained for $T_{RM} E_{RM}$.

When comparing our proposed entity-based methods with $T_{RM} E_{RM}$ we see that for some collections the former are statistically significantly better (WT10G, ClueBF) while for some collections (ROBUST, GOV2) the opposite holds. In the following we therefore try to evaluate the potential merits of using inter-entity similarity-based query models, specifically of models induced by utilizing clusters of similar entities.

Table 6.5: Comparison of entity-based query models with term-based query models. 'e', 't', '$*$', 'r', 's' and 'f' mark significant difference with E_Q, T_Q, E_{RM}, T_{RM}, T_QE_Q and $T_{RM}E_{RM}$, respectively.

	ROBUST			WT10G			GOV2			ClueBF		
	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG
E_Q	21.7	37.6	38.4	18.5	27.9	27.5	26.3	49.6	39.0	16.8	33.6	25.1
T_Q	25.2^{e}	42.7^{e}	43.9^{e}	19.0	28.0	30.9	29.5^{e}	53.1	44.9^{e}	17.9	32.1	22.9
E_{RM}	24.2^{e}	38.6^{t}	38.9^{t}	19.1	28.0	26.6	27.2^{e}	49.7	38.1^{t}	18.0^{e}	35.6	27.8^{e}
T_{RM}	$28.2^{e,t,*}$	$43.1^{e,*}$	$43.3^{e,*}$	19.3	28.8	30.5	$32.7^{e,t,*}$	$58.0^{e,t,*}$	$47.9^{e,t,*}$	19.2^{t}	34.7^{t}	25.5^{t}
T_QE_Q	$27.9^{e,t,*}$	$46.7^{e,t,*}_{r}$	$48.2^{e,t,*}_{r}$	$22.4^{e,t,*}_{r}$	$31.7^{e,t,*}$	$34.2^{e,*}_{r}$	$32.4^{e,t,*}$	$58.7^{e,t,*}$	$48.9^{e,t,*}$	$21.3^{e,t,*}_{r}$	$38.6^{e,t}_{r}$	28.5^{t}_{r}
T_QE_{RM}	$28.7^{e,t,*}_{s}$	$46.8^{e,t,*}_{r}$	$47.6^{e,t,*}_{r}$	$21.5^{e,t,*}$	$31.8^{e,t,*}$	$33.5^{e,*}$	$32.9^{e,t,*}$	$59.5^{e,t,*}$	$49.0^{e,t,*}$	$22.9^{e,t,*}_{r,s}$	$43.6^{e,t,*}_{r,s}$	$33.4^{e,t,*}_{r,s}$
$T_{RM}E_Q$	$29.7^{e,t,*}_{r,s}$	$47.5^{e,t,*}_{r}$	$48.2^{e,t,*}_{r}$	$22.8^{e,t,*}_{r}$	$31.5^{e,*}$	$33.5^{e,*}$	$34.7^{e,t,*}_{r,s}$	$\mathbf{62.6}^{\mathbf{e,t},*}_{\mathbf{r,s}}$	$52.0^{e,t,*}_{r,s}$	$21.8^{e,t,*}_{r}$	$39.5^{e,t}_{r}$	$29.7^{e,t}_{r}$
$T_{RM}E_{RM}$	$\mathbf{29.9}^{\mathbf{e,t},*}_{\mathbf{r,s}}$	$47.5^{e,t,*}_{r}$	$48.0^{e,t,*}_{r}$	$22.5^{e,t,*}_{r}$	$31.3^{e,*}$	$33.0^{e,*}$	$\mathbf{35.1}^{\mathbf{e,t},*}_{\mathbf{r,s}}$	$\mathbf{62.6}^{\mathbf{e,t},*}_{\mathbf{r,s}}$	$\mathbf{52.3}^{\mathbf{e,t},*}_{\mathbf{r,s}}$	$22.8^{e,t,*}_{r,s}$	$43.6^{e,t,*}_{r,s}$	$33.0^{e,t,*}_{r,s}$
$T_QE_{QS_r}(COS)$	$28.6^{e,t,*}_{s,f}$	$47.3^{e,t,*}_{r}$	$48.2^{e,t,*}_{r}$	$22.1^{e,t,*}_{r}$	$33.9^{e,t,*}_{r,f}$	$34.6^{e,*}$	$33.5^{e,t,*}_{s,f}$	$61.1^{e,t,*}$	$50.9^{e,t,*}$	$23.9^{e,t,*}_{r,s,f}$	$45.7^{e,t,*}_{r,s}$	$35.3^{e,t,*}_{r,s,f}$
$T_QE_{Clust_r}(COS)$	$28.4^{e,t,*}_{s,f}$	$\mathbf{47.6}^{\mathbf{e,t},*}_{\mathbf{r}}$	$\mathbf{48.8}^{\mathbf{e,t},*}_{\mathbf{r}}$	$\mathbf{23.8}^{\mathbf{e,t},*}_{\mathbf{r,s}}$	$\mathbf{34.5}^{\mathbf{e,t},*}_{\mathbf{r,s,f}}$	$\mathbf{35.9}^{\mathbf{e,t},*}_{\mathbf{r,f}}$	$32.7^{e,t,*}_{f}$	$60.4^{e,t,*}$	$50.3^{e,t,*}$	$23.7^{e,t,*}_{r,s,f}$	$45.0^{e,t,*}_{r,s}$	$34.3^{e,t,*}_{r,s}$
$T_QE_{QS_r}(QSSM1)$	$28.8^{e,t,*}_{s,f}$	$47.3^{e,t,*}_{r}$	$48.2^{e,t,*}_{r}$	$22.7^{e,t,*}_{r}$	$34.4^{e,t,*}_{r,s,f}$	$35.2^{e,t,*}_{r}$	$33.1^{e,t,*}_{f}$	$59.7^{e,t,*}$	$49.5^{e,t,*}_{f}$	$\mathbf{24.2}^{\mathbf{e,t},*}_{\mathbf{r,s,f}}$	$\mathbf{46.2}^{\mathbf{e,t},*}_{\mathbf{r,s}}$	$\mathbf{35.7}^{\mathbf{e,t},*}_{\mathbf{r,s,f}}$
$T_QE_{Clust_r}(QSSM1)$	$28.3^{e,t,*}_{s,f}$	$46.7^{e,t,*}_{r}$	$47.7^{e,t,*}_{r}$	$22.5^{e,t,*}_{r}$	$33.1^{e,t,*}_{r}$	$34.5^{e,*}_{r}$	$33.0^{e,t,*}_{s,f}$	$60.1^{e,t,*}$	$50.0^{e,t,*}$	$23.1^{e,t,*}_{r,s}$	$43.7^{e,t,*}_{r,s}$	$32.9^{e,t,*}_{r,s}$

Oracle experiments for the cluster-based query model induction method
To estimate to potential retrieval merits of utilizing entity clusters for query model induction we performed an 'Oracle' experiment where clusters are selected for inducing entity-based query models by their direct effect on retrieval performance.

Given a query q, a set of entity clusters $Cl()$ and a configuration of parameters values (λ from Equation 6.1, β from Equation 6.10, relevance model smoothing parameter and a set of similarity measure parameters), we performed the following. Entities constituting each cluster c in $Cl()$ were used to induced an entity-only query model as described in Section 6.4.3 (see Equation 6.8). Cluster c-based query model was then integrated with the entity-only query maximum likelihood estimate as described in Section 6.4.6. The induced entity-only query model was used, together with the term-only query model, for re-ranking documents in D_{init}, by calculating document scores as described in Section 6.4.7. The AP of the re-ranked list was recorded. For each query in the dataset and a configuration of parameters values, the cluster c in $Cl()$, yielding the best AP score, was selected. The final experiment result was calculated by selecting the configuration for which the best MAP score over queries in the dataset was obtained. This MAP score serves as the 'Oracle' test result.

Table 6.6 presents the results of the 'Oracle' experiments performed by utilizing various inter-entity similarity measures for inducing entity clusters. The baseline MAP score we compare with is that attained by optimizing the highly effective $T_{RM}E_{RM}$ retrieval method. Specifically, the configuration of parameter values, utilized by the $T_{RM}E_{RM}$ method, for which an optimal MAP score is obtained, was selected and fixed. The optimized MAP score serves as the baseline. We see that the retrieval performance numbers obtained by utilizing optimal entity clusters are substantially and statistically significantly better then that of the optimized baseline. This result attests to the effectiveness of using entity clusters

Table 6.6: Oracle experiment results: selecting the optimal cluster of similar entities for inducing entity-based query model. 'f' marks statistically significant difference with $T_{RM}E_{RM}$.

	ROBUST			WT10G			GOV2			ClueBF		
	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG	MAP	p@10	NDCG
$T_{RM}E_{RM}$	29.9	47.5	48.1	23.5	32.3	35.0	35.3	62.8	52.6	23.5	44.8	34.9
$T_QE_{Clust_r}(COS)$	37.7^f	62.7^f	64.5^f	32.3^f	49.9^f	50.4^f	40.4^f	75.6^f	61.0^f	32.0^f	60.3^f	49.7^f
$T_QE_{Clust_r}(QSSM1)$	37.3^f	62.4^f	64.4^f	31.7^f	48.0^f	48.9^f	40.3^f	75.8^f	60.7^f	31.6^f	59.5^f	48.8^f
$T_QE_{Clust_r}(QSSM3)$	37.7^f	62.7^f	64.5^f	32.5^f	49.2^f	49.9^f	40.5^f	75.8^f	61.1^f	32.0^f	60.3^f	49.7^f
$T_QE_{Clust_r}(OK)$	38.2^f	62.9^f	65.2^f	32.3^f	49.1^f	50.5^f	40.4^f	75.9^f	61.3^f	31.7^f	59.7^f	48.6^f
$T_QE_{Clust_r}(WLM)$	38.1^f	62.5^f	64.7^f	32.7^f	48.4^f	49.7^f	40.5^f	75.3^f	60.7^f	31.9^f	60.4^f	49.4^f
$T_QE_{Clust_r}(MI)$	38.3^f	63.9^f	65.5^f	31.9^f	48.6^f	49.4^f	40.4^f	76.0^f	61.4^f	31.7^f	60.8^f	49.1^f
$T_QE_{Clust_r}(WW2V)$	38.1^f	63.2^f	65.2^f	31.7^f	49.2^f	49.5^f	40.5^f	77.0^f	61.9^f	31.7^f	60.5^f	49.4^f
$T_QE_{Clust_r}(LWW2V)$	38.7^f	63.8^f	66.3^f	33.3^f	$\mathbf{50.3^f}$	51.5^f	41.1^f	77.0^f	61.6^f	32.1^f	$\mathbf{61.2^f}$	$\mathbf{49.8^f}$
$T_QE_{Clust_r}(TW2V)$	$\mathbf{39.2^f}$	$\mathbf{64.9^f}$	$\mathbf{66.6^f}$	33.4^f	50.1^f	$\mathbf{51.8^f}$	$\mathbf{41.4^f}$	$\mathbf{77.7^f}$	$\mathbf{62.2^f}$	$\mathbf{32.4^f}$	60.8^f	$\mathbf{49.8^f}$

for query model induction. Similarly to the second-order cluster hypothesis test results presented in Section 6.3.2, the measure which was found to be most effective is TW2V. A major challenge which remains open is how to find clusters that can be utilized for inducing effective query models.

In Figure 6.3 we present the optimized MAP scores, computed by selecting clusters for inducing query models in a decreasing order of their contribution to this score. Three different values of the β parameter (see Equation 6.10) were fixed while searching the optimal configuration, to examine the effect of query anchoring parameter value on retrieval performance. We see that for high values of β (i.e., 0.5, 0.9), between 10 and 20 clusters can be effectively used for inducing entity-based query models. This number emphasizes the merits of developing effective cluster ranking methods for inducing cluster-based query models. We note that our proposed ranking method, described in 6.4.3, was found to be effective according to the analysis presented above. However, the 'Oracle' experiment results show that the high potential in utilizing entity clusters is far from being fulfilled.

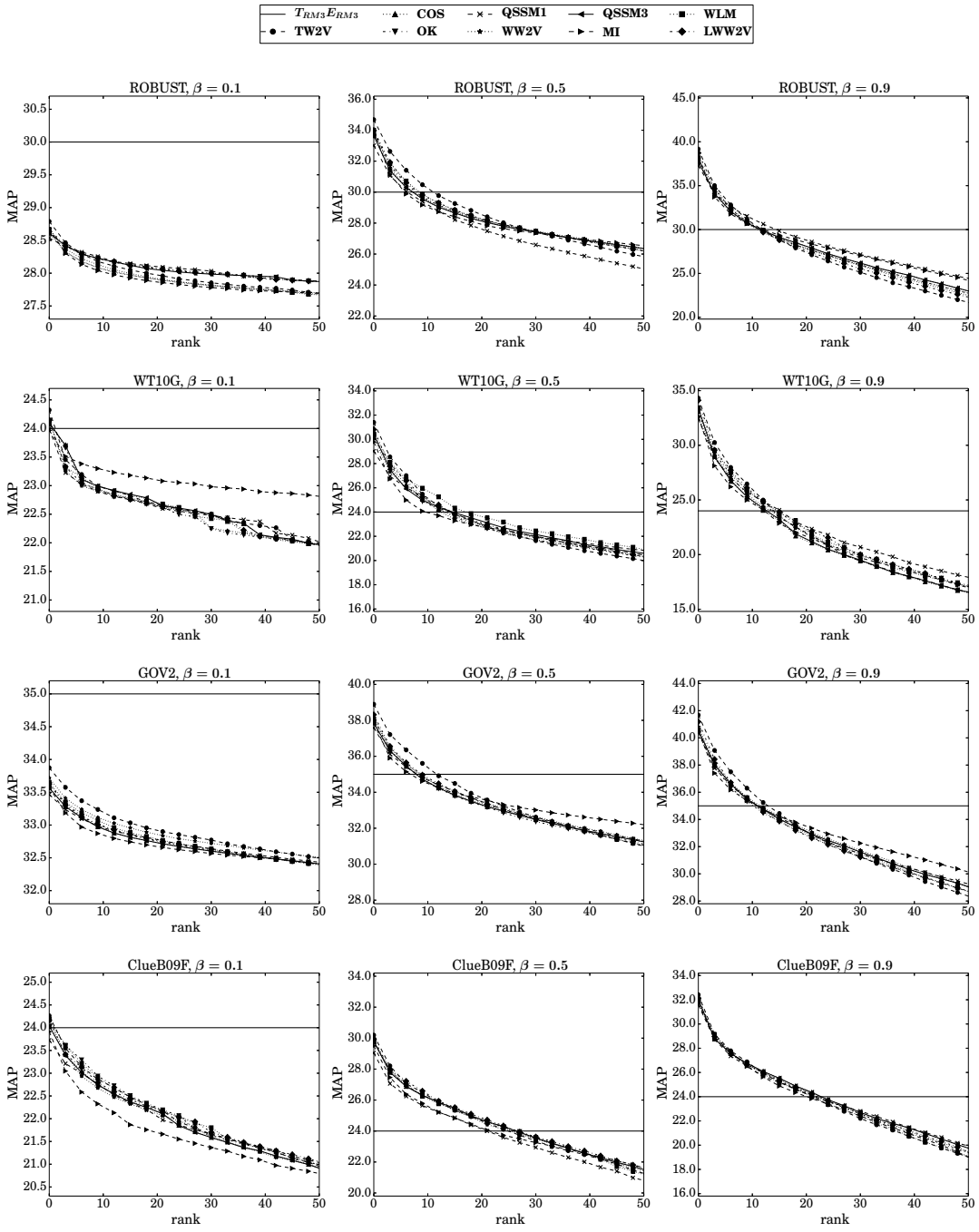

Figure 6.3: Best on average MAP score as a function of the cluster rank. Note: graphs are not to the same scale.

Chapter 7
Conclusions and Future Work

Our work has focused on methods for utilizing entities and entity-associated information for information retrieval. We addressed two different tasks: entity retrieval and ad hoc document retrieval.

Entity retrieval is the task of ranking entities with respect to a user's query [13, 15, 16, 22, 39, 44, 45, 47, 149]. We presented the first study of the cluster hypothesis for entity retrieval [144]. Voorhees' nearest-neighbor test [146] and several inter-entity similarity measures were utilized for testing the hypothesis. We showed that the hypothesis holds to a substantial extent for all the similarity measures we experimented with. In addition, the merits of using clusters of similar entities for entity ranking were demonstrated. We showed that ranking clusters according to the percentage of relevant entities they contain results in a highly effective entity ranking. Also, we proposed a method for ranking entity clusters with respect to a query and showed that this method improves retrieval effectiveness with respect to that of an effective initial search.

We also addressed the query performance prediction task for entity retrieval. We experimented with various types of predictors which were proposed for the task of ad hoc document retrieval [28], and showed that these predictors can be successfully adapted to the task of entity retrieval. In addition, we proposed a novel predictor that utilizes the retrieval scores of entity clusters to estimate retrieval effectiveness. The prediction quality of the proposed predictor was shown to be better or comparable to that of state-of-the-art predictors we experimented with.

Ad hoc document retrieval is the task of ranking documents with respect to a user query. To address a fundamental challenge regarding the use of entities for ad hoc document retrieval, we suggested novel entity-based query and document representations. These representations are language models which account, simultaneously, for terms and entity markups in the text, and for the uncertainty in the entity linking process. We showed that using these representations for document relevance estimation results in a highly effective retrieval. Specifically, our proposed methods outperformed state-of-the-art term proximity retrieval method [111, 77]. Also, we showed that the proposed representations can be effectively utilized for two additional retrieval paradigms: cluster-based ranking and query expansion.

Finally, we explored the merits of utilizing inter-entity similarities for document retrieval. A second-order cluster hypothesis for entities was proposed and evaluated. Specifically, we proposed a method for estimating entity relevance and used it, as well as various types of inter-entity similarity measures, to evaluated the hypothesis. In addition, we proposed several methods for inducing entity-based query models. We showed that retrieval methods utilizing our proposed query models are highly effective with respect to a few effective baselines. Finally, we showed that utilizing clusters of similar entities for query-model induction can result in extremely effective retrieval.

A few future work directions emerge as a result of our work. First, throughout this thesis we demonstrated the considerable potential of using clusters of similar entities for different retrieval tasks. The question of how to effectively rank entity clusters is still open and additional approaches can be explored and utilized. A natural research direction would be developing methods for learning to rank entity clusters.

Second, we proposed a method for estimating entity relevance with respect to a query. An effective estimation of entity relevance can be utilized in several ways, in both tasks that we addressed in this work: entity retrieval and document retrieval. For example, it would be interesting to evaluate the second-order cluster hypothesis for graded relevance judgments. In addition, it would be interesting to examine whether a user query can be addressed by both entities and documents that are relevant with respect to the underlying information need. Such approach poses a few interesting research questions. One of them is the question of how to effectively rank a list composed of both entities and documents with respect to a query.

Finally, it would be interesting to examine the merits of using entities in additional retrieval tasks. For example, entities may be helpful for estimating retrieval consistency, a challenge which has recently attracted research attention [5, 6]. Retrieval consistency is a new dimension in assessing the effectiveness of search systems - that is, how consistent are these systems in returning results in response to query variations that address the same information need [6]. Since many queries were found to be centered on entities [124], comparing entities in different query variations might be helpful for addressing this task. In a conceptually similar vein, entities can be utilized for the task of document lists fusion [3, 18, 50]. This task is aimed at creating a single ranked list of documents by fusing lists retrieved by different search systems. The different lists can be ranked with respect to their focus on entities that are relevant with respect to the information need. These ranks can then serve as an additional source of information in the fusion process.

BIBLIOGRAPHY

[1] N. Abdul-jaleel, J. Allan, W. B. Croft, O. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. Umass at TREC 2004: Novelty and hard. In *Proc. of TREC-13*, 2004.

[2] J. Allan, J. P. Callan, W. B. Croft, L. Ballesteros, J. Broglio, J. Xu, and H. Shu. Inquery at TREC-5. In *Proc. of TREC-5*, pages 119–132, 1996.

[3] Y. Anava, A. Shtok, O. Kurland, and E. Rabinovich. A probabilistic fusion framework. In *Proc. of CIKM*, pages 1463–1472, 2016.

[4] A. R. Aronson, T. C. Rindflesch, and A. C. Browne. Exploiting a large thesaurus for information retrieval. In *Proc. of RIAO*, pages 197–216, 1994.

[5] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. Uqv100: A test collection with query variability. In *Proc. of SIGIR*, pages 725–728, 2016.

[6] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. Retrieval consistency in the presence of query variations. In *Proc. of SIGIR*, pages 395–404, 2017.

[7] S. Balaneshin-kordan and A. Kotov. Sequential query expansion using concept graph. In *Proc. of CIKM*, pages 155–164, 2016.

[8] S. Balaneshinkordan and A. Kotov. An empirical comparison of term association and knowledge graphs for query expansion. In *Proc. of ECIR*, pages 761–767, 2016.

[9] K. Balog. *Encyclopedia of Database Systems*, chapter Entity Retrieval, pages 1–6. 2017.

[10] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proc. of SIGIR*, pages 43–50, 2006.

[11] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Information Processing & Management*, 45(1):1–19, 2009.

[12] K. Balog, M. Bron, and M. De Rijke. Query modeling for entity search based on terms, categories, and examples. *ACM Transactions on Information Systems*, 29(4):22:1–22:31, 2011.

[13] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the trec 2009 entity track. In *Proc. of TREC*, 2009.

[14] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Foundations and Trends in Information Retrieval.*, 6(2;3):127–256, 2012.

[15] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the trec 2010 entity track. In *Proc. of TREC*, 2010.

[16] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the trec 2011 entity track. In *Proc. of TREC*, 2011.

[17] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *Proc. of SIGIR*, pages 787–788, 2007.

[18] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Automatic combination of multiple ranked retrieval systems. In *Proc. of SIGIR*, pages 173–181, 1994.

[19] M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. In *Proc. of WSDM*, pages 31–40, 2010.

[20] M. Bendersky, D. Metzler, and W. B. Croft. Parameterized concept weighting in verbose queries. In *Proc. of SIGIR*, pages 605–614, 2011.

[21] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. T. Duc. Entity search evaluation over structured web data. In *Proc. of the 1st international workshop on entity oriented search (EOS), SIGIR*, 2011.

[22] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. Tran. Repeatable and reliable semantic search evaluation. *Web Semant.*, 21:14–29, Aug. 2013.

[23] W. C. Brandão, R. L. T. Santos, N. Ziviani, E. S. de Moura, and A. S. da Silva. Learning to expand queries using entities. *JASIST*, 65(9):1870–1883, 2014.

[24] S. Campinas, R. Delbru, N. A. Rakhmawati, D. Ceccarelli, and G. Tummarello. Sindice bm25mf at semsearch 2011. 2011.

[25] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 243–250, 2008.

[26] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at enterprise track of TREC 2005. In *Proc. of TREC*, volume 14, 2005.

[27] Y. Cao, J. Liu, S. Bao, H. Li, and N. Craswell. A two-stage model for expert search. Technical report, MSR-TR-2008, 2008.

[28] D. Carmel and E. Yom-Tov. Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, pages 1–89, 2010.

[29] J. Chen, C. Xiong, and J. Callan. An empirical study of learning to rank for entity search. In *Proc. of SIGIR*, pages 737–740, 2016.

[30] X. Cheng and D. Roth. Relational inference for wikification. In *Proc. of EMNLP*, pages 1787–1796, 2013.

[31] J. C. K. Cheung and X. Li. Sequence clustering and labeling for unsupervised query intent discovery. In *Proc. of WSDM*, pages 383–392, 2012.

[32] J. Chu-Carroll, G. A. Averboch, P. A. Duboue, D. Gondek, J. W. Murdock, J. M. Prager, P. Hoffmann, and J. Wiebe. Ibm in trec 2006 enterprise track. In *TREC*, 2006.

[33] M. Ciglan, K. Nørvåg, and L. Hluchý. The semsets model for ad-hoc semantic list search. In *Proc. of WWW*, pages 131–140, 2012.

[34] R. L. Cilibrasi and P. M. B. Vitanyi. The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):370–383, 2007.

[35] K. Collins-Thompson and J. Callan. Query expansion using random walk models. In *Proc. of CIKM*, pages 704–711, 2005.

[36] W. S. Cooper. Exploiting the maximum entropy principle to increase retrieval effectiveness. *Journal of the American Society for Information Science*, 34(1):31–39, 1983.

[37] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[38] M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *Proc. of WWW*, pages 249–260, 2013.

[39] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec 2005 enterprise track. In *Proc. of TREC*, 2005.

[40] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proc. of SIGIR*, pages 299–306, 2002.

[41] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proc. of EMNLP*, 2007.

[42] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proc. of SIGIR*, pages 365–374, 2014.

[43] J. Dalton and S. Huston. Semantic entity retrieval using web queries over structured rdf data. In *Proc. of the 3rd Intl. Semantic Search Workshop*, 2010.

[44] A. P. de Vries, A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas. Overview of the inex 2007 entity ranking track. In *Proc. of INEX*, pages 245–251, 2007.

[45] G. Demartini, T. Iofciu, and A. P. de Vries. Overview of the inex 2009 entity ranking track. In *Proc. of INEX*, pages 254–264, 2009.

[46] G. Demartini, P. Kärger, G. Papadakis, and P. Fankhauser. L3s research center at the sem-search 2010 evaluation for entity search track. In *Proc. of the 3rd Intl. Semantic Search Workshop*, 2010.

[47] G. Demartini, A. P. Vries, T. Iofciu, and J. Zhu. Advances in focused retrieval. chapter Overview of the INEX 2008 Entity Ranking Track, pages 243–252. 2009.

[48] F. Diaz. Performance prediction using spatial autocorrelation. In *Proc. of SIGIR*, pages 583–590, 2007.

[49] F. Diaz, B. Mitra, and N. Craswell. Query expansion with locally-trained word embeddings. *CoRR*, abs/1605.07891, 2016.

[50] M. Efron. Generative model-based metasearch for data fusion in information retrieval. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 153–162, 2009.

[51] O. Egozi, S. Markovitch, and E. Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2):8, 2011.

[52] A. El-Hamdouchi and P. Willett. Techniques for the measurement of clustering tendency in document retrieval systems. *Journal of Information Science*, 13:361–365, 1987.

[53] F. Ensan and E. Bagheri. Document retrieval model through semantic linking. In *Proc. of WSDM*, WSDM '17, pages 181–190, 2017.

[54] H. Fang and C. Zhai. Probabilistic models for expert finding. In *Advances in Information Retrieval*, pages 418–430. 2007.

[55] Y. Fang, L. Si, and A. P. Mathur. Discriminative models of integrating document evidence and document-candidate associations for expert search. In *Proc. of SIGIR*, pages 683–690, 2010.

[56] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with Wikipedia pages. *arXiv preprint arXiv:1006.3498*, 2010.

[57] P. Ferragina and U. Scaiella. Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proc. of CIKM*, pages 1625–1628, 2010.

[58] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. of ACL*, pages 363–370, 2005.

[59] J. F. Forst, A. Tombros, and T. Rölleke. Solving the enterprise trec task with probabilistic data models. In *TREC*, 2006.

[60] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proc. of AAAI*, pages 1301–1306, 2006.

[61] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proc. of IJCAI*, pages 1606–1611, 2007.

[62] D. Garigliotti and K. Balog. Towards an understanding of entity-oriented search intents. *arXiv preprint arXiv:1802.08010*, 2018.

[63] S. Gauch and J. Wang. A corpus analysis approach for automatic query expansion. In *Proc. of CIKM*, pages 278–284, 1997.

[64] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proc. of SIGIR*, pages 267–274, 2009.

[65] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. S. Thompson, and D. T. Tran. Evaluating ad-hoc object retrieval. In *Proc. of the Intl. Workshop on Evaluation of Semantic Technologies*, 2010.

[66] C. Hauff, D. Hiemstra, and F. de Jong. A survey of pre-retrieval query performance predictors. In *Proc. of CIKM*, pages 1419–1420, 2008.

[67] C. Y. B. P. J. He and Z. Yang. Cnds expert finding system for trec2005.

[68] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proc. of SIGIR*, pages 76–84, 1996.

[69] W. R. Hersh, D. H. Hickam, and T. Leone. Words, concepts, or both: optimal indexing units for automated information retrieval. In *Proc. of SCAMC*, page 644, 1992.

[70] D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *Proc. of SIGIR*, pages 35–41, 2002.

[71] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proc. of EMNLP*, pages 782–792, 2011.

[72] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Proc. of SIGIR*, pages 541–544, 2003.

[73] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *Proc. of SIGIR*, pages 179–186, 2008.

[74] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou. Exploiting Wikipedia as external knowledge for document clustering. In *Proc. of SIGKDD*, pages 389–396, 2009.

[75] A. Huang, D. Milne, E. Frank, and I. H. Witten. Clustering documents with active learning using Wikipedia. In *Proc. of ICDM*, pages 839–844, 2008.

[76] A. Huang, D. Milne, E. Frank, and I. H. Witten. Clustering documents using a Wikipedia-based concept representation. In *Proc. of PAKDD*, pages 628–636. 2009.

[77] S. Huston and W. B. Croft. A comparison of retrieval models using term dependencies. In *Proc. of CIKM*, pages 111–120, 2014.

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

[79] J. Jiang, W. Lu, X. Rong, and Y. Gao. Adapting language modeling methods for expert search to rank wikipedia entities. In *Advances in Focused Retrieval*, pages 264–272. 2009.

[80] Y. Jing and W. B. Croft. An association thesaurus for information retrieval. Technical report, Amherst, MA, USA, 1994.

[81] R. Kaptein and J. Kamps. Finding entities in wikipedia using links and categories. In *Advances in Focused Retrieval*, pages 273–279. 2009.

[82] R. Kaptein and J. Kamps. Exploiting the category structure of wikipedia for entity ranking. *Artificial Intelligence*, 0:111 – 129, 2013.

[83] R. Krovetz and W. B. Croft. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)*, 10(2):115–141, 1992.

[84] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proc. of KDD*, pages 457–466, 2009.

[85] G. Kumaran and J. Allan. A case for shorter queries, and helping users create them. In *Proc. of NAACL*, pages 220–227, 2007.

[86] H.-K. J. Kuo and W. Reichl. Phrase-based language models for speech recognition. In *Proc. of EUROSPEECH*, 1999.

[87] O. Kurland and C. Domshlak. A rank-aggregation approach to searching for optimal query-specific clusters. In *Proc. of SIGIR*, pages 547–554, 2008.

[88] O. Kurland and E. Krikon. The opposite of smoothing: A language model approach to ranking query-specific document clusters. *Journal of Artificial Intelligence Research (JAIR)*, 41:367–395, 2011.

[89] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proc. of SIGIR*, pages 194–201, 2004.

[90] O. Kurland and L. Lee. Pagerank without hyperlinks: Structural reranking using links induced by language models. *ACM Transactions on Information Systems (TOIS)*, 28(4):18, 2010.

[91] S. Kuzi, A. Shtok, and O. Kurland. Query expansion using word embeddings. In *Proc. of CIKM*, pages 1929–1932, 2016.

[92] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119, 2001.

[93] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of SIGIR*, pages 120–127, 2001.

[94] M. Levit, S. Parthasarathy, S. Chang, A. Stolcke, and B. Dumoulin. Word-phrase-entity language models: getting more mileage out of n-grams. In *Proc. of INTERSPEECH*, pages 666–670, 2014.

[95] R. Li, L. Hao, P. Zhang, D. Song, and Y. Hou. A query expansion approach using entity distribution based on markov random fields. In *Proc. of AIRS*, 2015.

[96] S. Liang and M. de Rijke. Finding knowledgeable groups in enterprise corpora. In *Proc. of SIGIR*, pages 1005–1008, 2013.

[97] T. Lin, P. Pantel, M. Gamon, A. Kannan, and A. Fuxman. Active objects: actions for entity-centric search. In *Proc. of WWW*, pages 589–598, 2012.

[98] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[99] X. Liu, F. Chen, H. Fang, and M. Wang. Exploiting entity relationship for query expansion in enterprise search. *Information Retrieval Journal*, 17(3):265–294, 2014.

[100] X. Liu and W. Croft. Experiments on retrieval of optimal clusters. Technical report, Technical Report IR-478, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts, 2006.

[101] X. Liu and W. B. Croft. Evaluating text representations for retrieval of the best group of documents. In *Proc. of ECIR*, pages 454–462, 2008.

[102] X. Liu and H. Fang. A study of entity search in semantic search workshop. In *Proc. of the 3rd Intl. Semantic Search Workshop*, 2010.

[103] X. Liu and H. Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.

[104] X. Liu, W. Zheng, and H. Fang. An exploration of ranking models and feedback method for related entity finding. *Information Processing & Management*, 49(5):995–1007, 2013.

[105] A. marie Vercoustre, J. Pehcevski, and V. Naumovski. Topic difficulty prediction in entity ranking. In *Proc. of INEX*, pages 280–291, 2009.

[106] O. Medelyan, D. Milne, C. Legg, and I. H. Witten. Mining meaning from wikipedia. *Int. J. Hum.-Comput. Stud.*, 67(9):716–754, 2009.

[107] E. Meij, K. Balog, and D. Odijk. Entity linking and retrieval. In *Proc. of SIGIR*, pages 1127–1127, 2013.

[108] E. Meij, L. Ijzereef, L. Azzopardi, J. Kamps, and M. de Rijke. Combining thesauri-based methods for biomedical retrieval. In *Proc. of TREC*, 2005.

[109] E. Meij, D. Trieschnigg, M. de Rijke, and W. Kraaij. Conceptual language models for domain-specific retrieval. *Information Processing & Management*, 46(4):448–469, 2010.

[110] E. Meij, W. Weerkamp, and M. De Rijke. Adding semantics to microblog posts. In *Proc. of WSDM*, pages 563–572, 2012.

[111] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proc. of SIGIR*, pages 472–479, 2005.

[112] D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. In *Proc. of SIGIR*, pages 311–318, 2007.

[113] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proc. of CIKM*, pages 233–242, 2007.

[114] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[115] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *Proc. of CIKM*, pages 509–518, 2008.

[116] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proc. of SIGIR*, pages 206–214, 1998.

[117] C. Moreira. Learning to rank academic experts. Master's thesis, Technical University of Lisbon, 2011.

[118] R. Neumayer, K. Balog, and K. Nørvåg. On the modeling of entities for ad-hoc entity search in the web of data. In *Advances in Information Retrieval*, pages 133–145. 2012.

[119] P. Ogilvie and J. Callan. Combining document representations for known-item search. In *Proc. of SIGIR*, pages 143–150, 2003.

[120] D. Pan, P. Zhang, J. Li, D. Song, J. Wen, Y. Hou, B. Hu, Y. Jia, and A. N. D. Roeck. Using Dempster-Shafer's evidence theory for query expansion based on freebase knowledge. In *Proc. of AIRS*, pages 121–132, 2013.

[121] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proc. of EMNLP*, pages 1532–1543, 2014.

[122] D. Petkova and W. B. Croft. Proximity-based document representation for named entity retrieval. In *Proc. of CIKM*, pages 731–740, 2007.

[123] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of SIGIR*, pages 275–281, 1998.

[124] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proc. of WWW*, pages 771–780, 2010.

[125] F. Raiber and O. Kurland. Ranking document clusters using markov random fields. In *Proc. of SIGIR*, pages 333–342, 2013.

[126] F. Raiber, O. Kurland, F. Radlinski, and M. Shokouhi. Learning asymmetric co-relevance. In *Proc. of ICTIR*, pages 281–290, 2015.

[127] H. Raviv, D. Carmel, and O. Kurland. A ranking framework for entity oriented search using markov random fields. In *Proc. of the 1st Joint International Workshop on Entity-Oriented and Semantic Search*, page 1, 2012.

[128] H. Raviv, O. Kurland, and D. Carmel. The cluster hypothesis for entity oriented search. In *Proc. of SIGIR*, pages 841–844, 2013.

[129] H. Raviv, O. Kurland, and D. Carmel. Document retrieval using entity-based language models. In *Proc. of SIGIR*, pages 65–74, 2016.

[130] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval.*, 3(4):333–389, Apr. 2009.

[131] S. E. Robertson. Readings in information retrieval. chapter The probability ranking principle in IR, pages 281–286. 1997.

[132] S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *Proc. of SIGIR*, pages 35–56, 1981.

[133] J. J. Rocchio. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, pages 313–323, 1971.

[134] D. Roy, D. Paul, M. Mitra, and U. Garain. Using word embeddings for automatic query expansion. *CoRR*, abs/1606.07608, 2016.

[135] G. Salton. A new comparison between conventional indexing (medlars) and automatic text processing (smart). *Journal of the American Society for Information Science*, 23(2):75–84, 1972.

[136] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[137] J. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[138] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems*, 30(2):11, 2012.

[139] M. D. Smucker and J. Allan. A new measure of the cluster hypothesis. In *Proc. of ICTIR*, pages 281–288, 2009.

[140] P. Srinivasan. Query expansion and medline. *Information Processing & Management*, 32(4):431–443, 1996.

[141] J. A. Thom, J. Pehcevski, and A.-M. Vercoustre. Use of wikipedia categories in entity ranking. *CoRR*, abs/0711.2917, 2007.

[142] A. Tombros and C. Van Rijsbergen. Query-sensitive similarity measures for information retrieval. *Knowledge and Information Systems*, 6(5):617–642, 2004.

[143] A. Tombros, R. Villa, and C. Van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing & management*, 38(4):559–582, 2002.

[144] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition, 1979.

[145] A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in wikipedia. In *Proc. of the 2008 ACM symposium on Applied computing*, pages 1101–1106, 2008.

[146] E. M. Voorhees. The cluster hypothesis revisited. In *Proc. of SIGIR*, pages 188–196, 1985.

[147] E. M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proc. of SIGIR*, pages 171–180, 1993.

[148] P. Wang and C. Domeniconi. Building semantic kernels for text classification using Wikipedia. In *Proc. of SIGKDD*, pages 713–721. ACM, 2008.

[149] Q. Wang, J. Kamps, G. Ramirez Camps, M. Marx, A. Schuth, M. Theobald, S. Gurajada, and A. Mishra. Overview of the inex 2013 linked data track. In *CEUR Workshop Proceedings*, volume 1179, 01 2012.

[150] Z. Wang, D. Liu, W. Xu, G. Chen, and J. Guo. Bupt at trec 2009: Entity track. In *In Proc. of TREC 2009*, 2009.

[151] Z. Wang, C. Tang, X. Sun, H. Ouyang, R. Lan, W. Xu, G. Chen, and J. Guo. Pris at trec 2010: Related entity finding task of entity track. Technical report, DTIC Document, 2010.

[152] J. S. Whissell and C. L. Clarke. Effective measures for inter-document similarity. In *Proc. of CIKM*, pages 1361–1370, 2013.

[153] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proc. of AAAI Workshop on Wikipedia and AI*, pages 25–30, 2008.

[154] Y. Wu and H. Kashioka. Nict at trec 2009: Employing three models for entity ranking track. In *TREC*. Citeseer, 2009.

[155] C. Xiong and J. Callan. EsdRank: Connecting query and documents through external semi-structured data. In *Proc. of CIKM*, pages 951–960, 2015.

[156] C. Xiong and J. Callan. Query expansion with Freebase. In *Proc. of ICTIR*, pages 111–120, 2015.

[157] C. Xiong, J. Callan, and T.-Y. Liu. Bag-of-entities representation for ranking. In *Proc. of ICTIR*, pages 181–184, 2016.

[158] C. Xiong, J. Callan, and T.-Y. Liu. Word-entity duet representations for document ranking. In *Proc. of SIGIR*, pages 763–772, 2017.

[159] C. Xiong, R. Power, and J. Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proc. of WWW*, pages 1271–1279, 2017.

[160] Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on Wikipedia. In *Proc. of SIGIR*, pages 59–66, 2009.

[161] X. Xue and X. Yin. Topic modeling for named entity queries. In *Proc. of CIKM*, pages 2009–2012, 2011.

[162] Y. Yang and C. G. Chute. Words or concepts: the features of indexing units and their optimal use in information retrieval. In *Proc. of SCAMC*, page 685, 1993.

[163] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating ap and ndcg. In *Proc. of SIGIR*, pages 603–610, 2008.

[164] X. Yin and S. Shah. Building taxonomy of web search intents for name entity queries. In *Proc. of WWW*, pages 1001–1010, 2010.

[165] G. You, Y. Lu, G. Li, and Y. Yin. Ricoh research at trec 2006: Enterprise track. In *TREC*, 2006.

[166] H. Zamani and W. B. Croft. Embedding-based query language models. In *Proc. of ICTIR*, pages 147–156, 2016.

[167] H. Zamani and W. B. Croft. Estimating embedding vectors for queries. In *Proc. of ICTIR*, pages 123–132, 2016.

[168] H. Zamani and W. B. Croft. Relevance-based word embedding. *arXiv preprint arXiv:1705.03556*, 2017.

[169] H. Zamani and W. B. Croft. Relevance-based word embedding. In *Proc. of SIGIR*, pages 505–514, 2017.

[170] C. Zhai. Statistical language models for information retrieval: A critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, 2008.

[171] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proc. of CIKM*, pages 403–410, 2001.

[172] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of SIGIR*, pages 334–342, 2001.

[173] Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proc. of ECIR*, pages 52–64, 2008.

[174] W. Zheng, S. Gottipati, J. Jiang, and H. Fang. Udel/smu at trec 2009 entity track. Technical report, DTIC Document, 2009.

[175] N. Zhiltsov and E. Agichtein. Improving entity search over linked data by modeling latent semantics. In *Proc. of CIKM*, pages 1253–1256, 2013.

[176] Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *Proc. of SIGIR*, pages 543–550, 2007.

[177] J. Zhu, D. Song, and S. Rüger. Integrating document features for entity ranking. In *Focused Access to XML Documents*, pages 336–347. 2008.

[178] J. Zhu, D. Song, and S. Rüger. Integrating multiple windows and document features for expert finding. *Journal of the American Society for Information Science and Technology*, 60(4):694–715, 2009.

הייצוג שאנו מציעים הוא חדשני משום שהוא לוקח בחשבון, באופן מקביל, מידע לגבי חוסר הודאות שבסימון הישויות ומידע לגבי המילים והישויות שהטקסט מכיל. אנו משתמשים ביצוגים המוצעים על מנת לייצג מסמכים ושאילתות ומראים שהם יכולים לשמש בתהליך האחזור. בסדרת ניסויים אנו מדגימים כי איכות האחזור המתקבלת כתוצאה משימוש בייצוגים אלו היא גבוהה וטובה יותר מהאיכות המתקבלת כתוצאה משימוש בשיטות הנחשבות אפקטיביות. לדוגמא, השיטה שאנו מציעים אפקטיבית יותר משיטה מובילה העושה שימוש במרחק בין מילות שאילתא שנמצאות במסמך על מנת להעריך את הרלוונטיות שלו לצורך במידע.

בשלב שני אנו מציגים שיטות חדשות ליצירת מודלי שאילתא שמייצגים את הצורך במידע באמצעות ישויות. שאילתות שמופנות למנועי חיפוש הן בדרך כלל קצרות ולא מייצגות היטב את הצורך במידע של המשתמש. אנו מציעים לעשות שימוש במידע ממוקד הקשור לישויות והוא הקשרים הסמנטים ביניהן, על מנת ליצור מודל עשיר יותר של הצורך במידע. כדוגמא, הישויות הקרובות ביותר מבחינה סמנטית לישויות שבשאילתא מקבלות ציון גבוה באחד ממודלי השאילתא שאנו מציגים, משום שהן מוערכות כחשובות ביחס לצורך מידע. מעבר להצגת מודלי השאילתא החדשים אנו מבצעים ניתוח איכותי של פוטנציאל השימוש בקשרים בין ישויות עבור משימת האחזור. אנו מציעים את הפותזת הקלסטר עבור ישויות הקשורות לצורך במידע ומראים שישויות שדומות אחת לשניה הן גם רלוונטיות עבור אותו הצורך במידע.

על מנת לחקור את השיטות המוצעות ליצירת מודלי שאילתא אנו מבצעים סדרת ניסויים שעושה שימוש בצורות הערכה שונות של קשרים סמנטיים בין ישויות. בנוסף אנו מציעים שיטות מגוונות ליצירת מודלי שאילתא. אנו מראים שבאמצעות השיטות שהצענו ניתן ליצור מודלי שאילתא שהם אפקטיבים לאחזור מסמכים. איכות האחזור המתקבלת גבוהה יותר מאשר זו המתקבלת על ידי שימוש בשאילתא שנמסרה על ידי המשתמש לבדה.

אחד האתגרים המשמעותיים ביותר בתחום אחזור המידע הוא הצורך לזהות באופן אפקטיבי מידע שעונה לצורכי המשתמשים, המנוסחים על ידי שאילתות. בעידן הנוכחי, משתמשים מצפים לתשובות קצרות וממוקדות במענה לחיפוש שביצעו. הדבר נכון במיוחד כאשר מדובר בחיפוש שמבוצע ממכשירים ניידים בעלי גודל מסך קטן.

ישויות הן פרטי מידע בעלי משמעות סמנטית, למשל, שמות אנשים, מקומות, ארגונים וכד'. בדרך כלל, ישויות מאוגדות במאגרי מידע כגון ויקיפדיה.

בשנים האחרונות בוצעו מחקרים שונים שהראו שמרבית צרכי המידע של משתמשים ממוקדים ביישויות. ניתוח שאילתות שהתקבלו על ידי מנועי חיפוש שונים הראה כי רוב השאילתות מכילות ישויות ושהצורך במידע של המשתמשים קשור ישירות ליישויות אלו. הממצאים הללו הובילו לעניין הולך וגובר בשימוש בישויות על מנת לענות על צרכי המידע של משתמשים.

שימוש אחד בישויות בתחום אחזור המידע מבוסס על "החלפת" פרטי המידע המאוחזרים ממסמכים לישויות. משימת מחקר שהוגדרה ונחקרה רבות בעשורים האחרונים היא משימת אחזור הישויות. מטרת משימה זו לענות על צרכי מידע שממוקדים בישויות, ושיכולים להענות על ידי אחזור ישויות במקום מסמכים.

שימוש אחר בישויות בתחום אחזור המידע מבוסס על שימוש במידע הסמנטי העשיר המלווה לישויות על מנת לזהות באופן אפקטיבי מסמכים שעונים לצורכי המידע של משתמשים. משימת אחזור המסמכים היא משימה בסיסית וחשובה בתחום אחזור המידע שנחקרת מזה שנים רבות.
שיטות רבות שהוצעו על מנת להתמודד עם משימה זו עושות שימוש בייצוגים טקסטואליים המבוססים על מילים שהטקסט מכיל. הבעיה ביצוגים מסוג זה היא שהמשמעות הסמנטית של המילים לא מיוצגת היטב. ההנחה בבסיס הגישה המבקשת לעשות שימוש בישויות על מנת לאחזר מסמכים, היא שהמשמעות הסמנטית שהן מיצגות תוכל לסייע במשימת האחזור.

שאלת המחקר העיקרית שאנו מציגים בעבודה זו היא: כיצד ניתן להשתמש בישויות על מנת לענות על צרכי המידע של משתמשים?

אנו עונים על שאלת המחקר על ידי שימוש בשתי הגישות שהוצגו לעיל.

ראשית, אנו עוסקים במשימת אחזור הישויות. המשימה הספציפית בה אנו מתמקדים היא משימת חיפוש יישויות במאגר ישויות גדול - ויקיפדיה. בעבודתינו אנו מציגים את המחקר הראשון של היפותזת הקלסטר, היפותזה מוכרת בעולם אחזור המסמכים, עבור משימת אחזור הישויות. אנו מראים שישויות שדומות זו לזו הן רלוונטיות לאותו צורך במידע. בהתבסס על תוצאות אלו אנו מציעים שיטה חדשה לאחזור ישויות שמבוססת על יצירת קבוצות של ישויות שדומות זו לזו. קבוצות אלו מדורגות על פי קשריהן לצורך במידע שמציגה השאילתא ודירוג זה מתורגם לדירוג של ישויות. אנו מראים שלשיטה זו יש פוטנציאל רב לביצוע משימת אחזור הישויות באופן אפקטיבי.

בנוסף, אנו מציגים מחקר שמטרתו להעריך, ללא שיפוטי רלוונטיות, את איכות החיפוש שבוצע על פני מאגר ישויות. אנו משתמשים בשיטות שונות שהוצעו עבור משימת אחזור המסמכים על מנת להעריך את איכות החיפוש, ומראים ששיטות אלו תקפות גם למשימת אחזור הישויות. בנוסף, אנו מציעים שיטה חדשה לשערוך איכות האחזור ומראים שהיא אפקטיבית ביחס לשיטות שהוצעו בעבר.

שנית, אנו עוסקים במשימת אחזור המסמכים. לצורך ההתמודדות עם משימה זו אנו עושים שימוש בטכנולוגיה שפותחה בשנים האחרונות ומטרתה לסמן ישויות בטקסטים מסוגים שונים. סימון של ישות בטקסט מכיל מזהה ייחודי של הישות ובנוסף מספר בין 0 ל 1 שמייצג את מידת הוודאות בסימון זה.

בשלב ראשון אנו מציגים שיטה חדשה לייצוג טקסט שמבוססת על ישויות ועל המילים הבסיסיות שהוא מכיל. שיטה זו מבוססת על מודלי שפה בהם משתמשים בעולם אחזור המידע.

הבעות תודה

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

המחקר נעשה בהנחיתו של פרופסור חבר אורן קורלנד בפקולטה
להנדסת תעשיה וניהול, טכניון – המכון הטכנולוגי לישראל

אחזור מבוסס ישויות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר דוקטור לפילוסופיה

הדס רביב

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
חיפה ניסן תשע"ח מרץ 2018