Information Retrieval manuscript No.
(will be inserted by the editor)

Latent Entity Space: A Novel Retrieval Approach for Entity-Bearing Queries

Xitong Liu · Hui Fang

Received: date / Accepted: date

Abstract Analysis on Web search query logs has revealed that there is a large portion of entity-bearing queries, reflecting the increasing demand of users on retrieving relevant information about entities such as persons, organizations, products, etc. In the meantime, significant progress has been made in Web-scale information extraction, which enables efficient entity extraction from free text. Since an entity is expected to capture the semantic content of documents and queries more accurately than a term, it would be interesting to study whether leveraging the information about entities can improve the retrieval accuracy for entity-bearing queries.

In this paper, we propose a novel retrieval approach, i.e., Latent Entity Space (LES), which models the relevance by leveraging entity profiles to represent semantic content of documents and queries. In the LES, each entity corresponds to one dimension, representing one semantic relevance aspect. We propose a formal probabilistic framework to model the relevance in the high-dimensional entity space. Experimental results over TREC collections show that the proposed LES approach is effective in capturing latent semantic content and can significantly improve the search accuracy of several state-of-the-art retrieval models for entity-bearing queries.

Keywords latent entity space · entity profile · document retrieval

X. Liu · H. Fang
Department of Electrical and Computer Engineering, University of Delaware
Newark, DE 19716, USA
E-mail: xtliu@udel.edu

H. Fang
E-mail: hfang@udel.edu

1 Introduction

The boom of Web technology yields the dramatic increase of data published in the recent decade, and it has been a long-standing challenge to develop effective Information Retrieval (IR) models to help users access relevant information. Traditional IR models (e.g., vector space models [46], classical probabilistic retrieval models [45], language modeling approaches [43]) assume that terms in queries and documents are independent and model the relevance based on the bag-of-words representations of queries and documents, making it possible to favor non-relevant documents with more occurrences of query terms.

Search has moved beyond the term-based document retrieval paradigm in recent years, as there is an increasing portion of Web search queries bearing entities [44]. Lin et al. [35] revealed that about 43% of the queries issued to one major commercial Web search engine contain entities. Moreover, a substantial portion of Web documents mention entities, and the advances in Web-scale information extraction make it possible to efficiently identify entities mentioned in the Web documents [4, 9, 19]. Since an entity is a better semantic unit than a term, it would be interesting to study how to leverage the entity information to better model the relevance between entity-bearing queries and documents.

Let us consider an entity-bearing query "discussion of the impending sale of the rocky mountain news". The query contains a named entity, i.e., *Rocky Mountain News*, which was a daily newspaper published in Denver, Colorado until February 27, 2009. Figure 1(a) shows a document about *Rocky Mountain*, and Figure 1(b) shows a document about *Rocky Mountain News*. It is clear that the second document is relevant while the first one is not. However, traditional retrieval models would favor the first document since it matches more occurrences of query terms.

In this paper, we propose a novel retrieval approach, i.e., Latent Entity Space (LES), which models the relevance between queries and documents through latent entities. The key idea is to construct a high-dimensional latent entity space, in which each dimension corresponds to one entity, and map both queries and documents to the latent space accordingly. The relevance between query and document is then estimated based on their projections to each dimension in the latent space. This is in contrast to the traditional term-based retrieval models, which estimate the query-document relevance in a high-dimensional term space. The main advantage of the entity-based space over the term-based space is that entities can capture the semantic content of documents and queries much better than terms.

As shown in Figure 1(b), the existence of query entity (i.e., *Rocky Mountain News*) and other useful entities (e.g., *Denver, Colorado, E.W. Scripps Co.*[1] and *The Denver Post*[2]) implies that this document is more likely to be relevant. Clearly, information about these entities should be considered as an

[1] A media group which owned Rocky Mountain News.

[2] A daily newspaper which is the rival of Rocky Mountain News in Denver, Colorado.

Rocky Mountains

The Rocky Mountains are a broad mountain range in western North America. The range's northernmost point is in British Columbia, Canada and its southernmost point is in New Mexico.

...

Rocky Mountain News

• Google News: Rocky Mountains

• Yahoo! News Search: Rocky Mountains

...

Rocky Mountain Blogs and Forums

• Rocky Mountain Nature Photographers: Discussion Forums

...

(a) non-relevant

Contact Sen. Ken Salazar

Nearly as long as there has been a *Denver, Colorado*, there has been a Rocky Mountain News. Help us save the Rocky.

Dear Sen. Salazar:

The *E.W. Scripps Co.* has announced its plan to sell the Rocky Mountain News. Unless a buyer emerges, *Colorado*'s longest running business may see its doors close after 150 years, leaving more than 200 tax-paying, voting Coloradans out of work.

...

Please work to ensure that the dissolution of the joint operating agreement that governs the partnership of *The Denver Post* and the Rocky Mountain News follows both the spirit and the letter of the law ...

(b) relevant

Fig. 1: Excerpts of two documents for query "*discussion of the impending sale of the rocky mountain news*". Matched query terms are underlined and other useful entities are in *italic*.

important semantic aspect in relevance modeling. Through projecting documents to the dimensions of these entities, LES is capable of capturing such semantic relevance.

A major challenge in LES is how to represent the information for each dimension, i.e., entity. A simple way would be to use the entity name but this is unlikely to work well since it can not represent much information about the entity. Thus, we propose to represent each dimension with the profile of the corresponding entity and explore two different strategies to estimate the entity profile. The first method is based on the information from the document collection. Information about entities is often scattered in multiple documents, so we propose to pool pieces of information from documents mentioning the

entities to restore the complete picture. Alternatively, thanks to the contributions of online community, a handful of user generated knowledge bases (e.g., DBpedia, Freebase, Wikipedia) have been well curated and become publicly available, and they provide much richer information about entities than documents. Thus, the second method is to leverage such online knowledge bases to construct the entity profile.

To make the retrieval model more effective and efficient, LES is not constructed based on all entities. Instead, it is query-dependent. For each query, only a few latent entities that are most related to the query are selected to construct LES. Once the dimensions of LES have been identified, the relevance score of a document for a query is then estimated based on both the projections of the document and the query to LES.

We conduct experiments over the TREC ClueWeb09 collection with Freebase annotations [30]. Experimental results in Section 5 show that LES can deliver significant improvements when combined with several state-of-the-art retrieval methods for entity-bearing queries, demonstrating the capability of LES on capturing additional semantic content that can not be captured by existing methods such as Relevance Model [34], Latent Concept Expansion [40]. Besides, we are aware that Dalton et al. [20] proposed an Entity Query Feature Expansion (EQFE) model which enriches the query with various entity related features (e.g., related entities, categories, Wikipedia, entity context, collection feedback, etc.) and conduct the experiment on exactly the same ClueWeb09 collection with the same Freebase annotation [30]. We conduct side by side comparison in Section 6.1 between EQFE and LES and demonstrate that LES outperforms EQFE significantly and is more robust against the low quality of entity annotation. Lastly, we conduct extensive evaluation for LES on TREC 2013 Web track test collection, providing additional evidence in favor of the effectiveness of LES.

We make the following contributions:

1. We propose a novel retrieval framework which can capture the latent semantic relations between queries and documents through entities.
2. We propose to estimate the entity profile from document collection directly, even in the absence of knowledge base annotations.
3. We extensively evaluate our LES based models on several standard datasets from TREC 2009 to TREC 2013 Web tack under different experimental settings, and demonstrate that our proposed LES model could deliver superior performance than several state-of-the-art methods based on side-by-side comparison.

The rest of the paper is organized as follows. Section 3 describes the problem formulation, and Section 4 provides the details about our proposed approach. Experiments results are reported and analyzed in Section 5 and Section 6. We then discuss the related work in Section 2. Finally, we conclude in Section 8.

2 Related Work

2.1 Concept-based IR

Due to the use of "bag of words" representation for both queries and documents, traditional IR models have the limitation of retrieving only syntactically relevant but not semantically relevant documents as well as missing some relevant documents with no explicit term match with the query. Concept-based IR was proposed to overcome the limitation of keyword-based approach. Query and document are both represented in high-level semantic concepts, and the relevance between them is estimated in concept-space, making it capable of capturing semantic correlations even in the presence of vocabulary gap. Vallet et al. [49] proposed an ontology-based retrieval model which encodes queries to weighted concept vectors and performs implicit query expansion based on class hierarchies in ontology. A set of tuples are retrieved by the vector based queries and documents are selected based on their semantic annotations and their corresponding mapping to the retrieved tuples. Styltsvig [48] studied how to utilize conceptual knowledge in ontology to improve retrieval performance. Shallow natural language processing is employed to map documents to concepts in the index phrase, and similarities between concepts are estimated based on several ontological features like structural distance, which ultimately serve for the estimation of query-document relevance. Grootjen et al. [31] explored a hybrid approach to perform query expansion by conducting formal concept analysis from the results of initial retrieval with the help of global thesauri-like information from corpus. Bendersky and Croft [5] studied how to extract key concepts in verbose queries and integrate them into a probabilistic model to improve effectiveness. More recently, Egozi et al. [25] proposed to employ Explicit Semantic Analysis [29] to augment the bag-of-words representation of queries and documents with comprehensive explicit concepts from Wikipedia as new text features in both indexing and retrieval phases, and apply self-generated labeled training data for effective feature selection. Different from existing concept-based IR approaches where queries or documents are transferred to concept based representation, LES does not alter the representation of queries and documents, and no explicit document-concept mapping is performed either. Instead, it uses entity profiles as bridge to measure semantic relevance between queries and documents in their intact representation. Besides, since the relevance is estimated based on projection in a general framework, it is more flexible to subsume existing language modeling approaches or other knowledge base related features for projection estimation in LES. Moreover, existing methods like Explicit Semantic Analysis [29] relies on the statistical inference on high-quality knowledge base like Wikipedia, while the effectiveness of collection based entity profile across multiple collections proves that LES could work well even in the absence of textual data in knowledge base. This is particularly useful for serving time-sensitive information needs where rich and up-to-date textual data may not be available for

the related entities in knowledge base like Wikipedia due to notable editorial time lag in the population process [28].

Another commonly used approach to concept-based IR is topic modeling, which aims to capture the relationships between terms through grouping them into topics automatically based on global and local statistics. Notable approaches include Latent Semantic Indexing [21] and Latent Dirichlet Allocation [7] based document modeling [51], and noticeable improvements could be observed. Nevertheless, topic modeling is often computationally expensive and the generated concepts are often difficult to interpret, making the application to large scale (e.g., the Web) infeasible and untraceable. In contrast, the entity model in LES could be obtained offline and relative low computation cost and is well suitable for parallel execution, and does not require complicated statistical inference as topic modeling does, thus clearly is more applicable to Web scale data. Besides, LES exploits explicit entities from human-curated knowledge base rather than implicit topics with no prior knowledge to represent the semantic aspects for query and document, making it more easy to interpret and analysis.

2.2 Entity Retrieval

Our work is related to entity retrieval, as LES needs to select top-k related entities with regard to query serve as dimensions in LES, a process similar to entity retrieval where entities matching the relevance criterion (e.g., relation, type, etc.) in query will be returned to fulfill user's information need.

Entity retrieval was first investigated in the expert search task of the Enterprise Track [18,47] in the TREC conference with the goal to find people with specific expertise. As preliminary endeavor, only one entity type (i.e., person) was studied though, it received intensive attention from the research community. Notable approaches include leverage documents to model the expert's knowledge [1], probabilistic generative models using documents to connect queries and experts [27], voting based on data fusion across a range of document and field weighting models [38] and proximity based document modeling [42]. It is interesting to note that Demartini et al. [24,22] proposed a novel vector space model to rank the expert entities by representing queries and entities in the vector space of topics, where each topic serves as one dimension. The relevance score between the query and entity is estimated by cosine similarity between their vector representations. Although the topic vector space and LES are trying to solve different IR problems, they shares some commonalities in the sense that they both leverage *latent space* to model relevance.

The entity track of the TREC conference [2,3] continued the efforts by generalizing the entity types from person to other types and extending the keyword query to structured query including input entity, type of target entities and relation between input and target entities, and the relation is the central part of query. On the other hand, the Entity Ranking track of INEX [23] also studied the problem of entity retrieval but emphasized more on the type of

target entities rather than the relation. Unfortunately, existing entity retrieval models can not be directly applied in LES to select top relevant entities for a query because the LES needs to select entities reflecting certain semantic aspects of the query and no explicit information about the entity type or the relations is provided in the query.

2.3 Leveraging Knowledge Bases

The public availability of well curated knowledge bases (e.g., Wikipedia, DBpedia, Freebase) makes people access structured information about entities in a more comprehensive way, and much richer information provided by knowledge bases makes it possible to be leveraged to improved document retrieval. Milne et al. [41] devised Koru, a Wikipedia back-ended search engine which could perform thesaurus-based automatic query expansion by utilizing a concept graph in Wikipedia and serve domain-independent exploratory queries very well. Elsas et al. [26] proposed a novel query expansion model by using the link structure in Wikipedia and it could improve performance significantly and consistently for blog feed retrieval task. Xu et al. [52] mapped queries to Wikipedia entity pages to represent the underlying knowledge of the query and expanded the queries with the information from the mapped Wikipedia articles. Liu et al. [36] explored how to leverage related entities of query and their relationship to perform query expansion based on both documents and DBpedia. Instead of performing query expansion, LES leverages the information from knowledge bases to build entity profiles and uses them to model the query-document relevance indirectly, therefore could avoid the common expansion-specific problems like query drift and high parameter sensitivity [6] and deliver more effective and robust performance.

More recently, Dalton et al. [20] proposed Entity Query Feature Expansion (EQFE) model which leverages various collection based and knowledge based features to improve retrieval effectiveness. Instead of performing traditional query expansion on the text representation, EQFE extends query with various features including entity name, entity links and attributes in knowledge base, entity context model, collection feedback, etc. The final query-document relevance is based on the integration of relevance between document and all the query features. Although this work shares some similarities with LES in terms of problem setup, there are some fundamental differences between them. First, EQFE uses enriched features to aggregate relevance between query and documents, while LES uses entity profiles to aggregate relevance. Second, EQFE requires learning-to-rank based parameter tuning to acquire parameters as the number of parameters to proportional to the number of features (at least 42), which is complicated and is vulnerable to overfitting, while LES uses fewer parameters and is more robust to parameter settings. Besides, EQFE requires explicit entity annotation in relevant documents to improve the performance as some important features rely on such annotations. In contrast, LES does not necessarily require entity annotation in relevant documents as the entity mod-

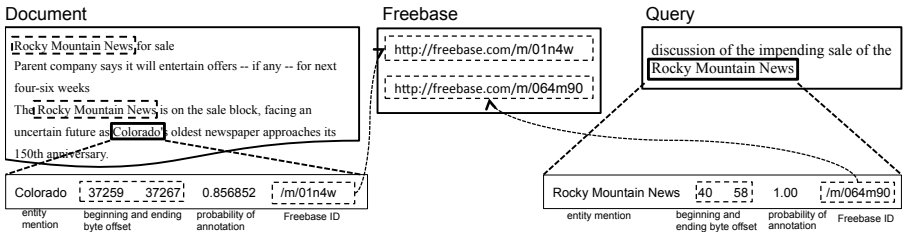


Fig. 2: Example Freebase annotations on ClueWeb09 (Note: not all entity annotations are displayed).

els are estimated from the whole collection or knowledge base and therefore is more robust against the low quality of entity annotation on partial documents. Experimental results in Section 6.1 confirm that LES could outperform EQFE significantly on side-by-side comparison.

3 Problem Formulation

The basic problem setup is the same as classic ad hoc information retrieval: given a keyword query q and a document collection $\mathcal{D} = \{d_1, d_2, \cdots, d_N\}$, we need to retrieve a list of documents ranked by their relevance with regard to q.

In addition to the queries and documents, we assume that we have entity annotations provided for both queries and documents. These annotations can be generated by employing existing Web-scale entity extraction methods. Instead of generating the data by ourselves, we choose the ClueWeb09 collection with Freebase annotations [30] in our study. Example entity annotations from the dataset for both a query and a document are shown in Figure 2.

We now explain the notations used in the rest of this paper. $\mathcal{E}$ denotes the entity space, $\mathcal{K}$ denotes a knowledge base which contains entries of entities in $\mathcal{E}$. $E(d)$ denotes the set of entity annotations in d, and $E(q)$ denotes the set of entity annotations in q. For each entity $e \in E(d)$, a set of meta information items are provided in the Freebase annotation dataset:

- $m(e)$: entity mention, i.e., the surface name of e in d. (e.g., "Colorado" in the example document in Figure 2.)
- $pos(e)$: the position of $m(e)$ in d. Note that $pos(e)$ is offset to the center of $m(e)$ by *term*, and it can be derived from the *byte* offset provided in entity annotation (e.g., the second and third column of annotation (37259 and 37267 respectively) in Figure 2).
- $p\big(e|m(e), d\big)$: the posterior probability of identifying e given both entity mention $m(e)$ and context in d, which refers to the fourth column (i.e., 0.856852) of annotation in Figure 2.

– $kb(e)$: the entry of e in $\mathcal{K}$, which would be a document carrying information about e. The fifth column of entity annotation in Figure 2 is the ID of the entry in Freebase, through which the whole entry can be accessed.

Note that all the above information is provided in the annotated ClueWeb09 collection [30], and similar information is also available for entities in the query, i.e., $e \in E(q)$.

4 Latent Entity Space

4.1 The Language Modeling Approach

Before we discuss the Latent Entity Space framework, let us briefly review the probabilistic models for document ranking.

The generative relevance modeling [33] provides a fundamental principle for language modeling approach to model query-document relevance. The basic idea is to estimate the relevance of document d with respect to query q based on the probability $p(\mathcal{R}=1|q,d)$, where $\mathcal{R}$ is a binary random variable denoting the relevance. By applying Bayes' rule, we get the log-odds ratio, a probabilistic equivalent for the ranking of documents:

$$p(\mathcal{R}=1|q,d) \overset{\text{rank}}{=} \log \frac{p(q,d|\mathcal{R}=1)p(\mathcal{R}=1)}{p(q,d|\mathcal{R}=0)p(\mathcal{R}=0)} \tag{1}$$

where $\overset{\text{rank}}{=}$ means the two values are equivalent with regard to the ranking of d. By assuming the query is generated by a probabilistic model based on the document, the conditional probability in Equation (1) can be factored as follows:

$$\begin{aligned} p(\mathcal{R}=1|q,d) &= \log \frac{p(q|d,\mathcal{R}=1)p(d|\mathcal{R}=1)p(\mathcal{R}=1)}{p(q|d,\mathcal{R}=0)p(d|\mathcal{R}=0)p(R=0)} \\ &= \log \frac{p(q|d,\mathcal{R}=1)}{p(q|d,\mathcal{R}=0)} + \log \frac{p(\mathcal{R}=1|d)}{p(\mathcal{R}=0|d)}. \end{aligned} \tag{2}$$

By assuming d is independent of q conditioned on the event $\mathcal{R}=0$, d and $\mathcal{R}$ are independent (i.e., no prior knowledge about the relevance of d), we obtain:

$$\begin{aligned} p(\mathcal{R}=1|q,d) &= \log \frac{p(q|d,\mathcal{R}=1)}{p(q|\mathcal{R}=0)} + \log \frac{p(\mathcal{R}=1)}{p(\mathcal{R}=0)} \\ &\overset{\text{rank}}{=} \log p(q|d,\mathcal{R}=1) \\ &\overset{\text{rank}}{=} \prod_{w \in q} p(w|\theta_d,\mathcal{R}=1)^{n(w,q)} \end{aligned} \tag{3}$$

where $p(q|d,\mathcal{R}=1)$ is the query likelihood, θ_d is a language model estimated from document d. This is known as the language modeling (LM) approach [43]. $n(w,q)$ denotes the number of occurrences of w in q.

4.2 Formal Derivation

Let us revisit the derivation of Equation (2). The underlying assumption is that q is generated by a probabilistic model based on d, implying q and d are connected through a probabilistic model θ_d (which is a term-based probabilistic distribution over the vocabulary) in Equation (3). The query likelihood $p(q|d, \mathcal{R} = 1)$ is essentially estimated directly in a high-dimensional term-space (i.e., the vocabulary) in which each term represents one dimension of relevance.

Due to the existence of *polysemy* (which is a common phenomenon in English), one term may have several semantic aspects, which makes it possible that a query and a document may have a high similarity in the term-based space but actually deviate from each other semantically. Moreover, multiple terms may share the same meaning (e.g., *synonymy*), but they are presented in different dimensions in the term-based space. It may therefore be inaccurate to capture the relevance in the term-based space. An entity, on the other hand, is a better alternative to a term with the following reasons:

- An entity is an atomic semantic concept, thus mitigating the problem of polysemy. Although distinct entities may share the same surface name, they are disambiguated and uniquely identified in existing knowledge bases. For example, Apple_(technology_company) and Apple_(fruit) are the unique IDs in Wikipedia for term "apple".
- An entity profile is the collection of its semantic aspects. A complete entity profile should include everything about the entity. For example, through the Wikipedia page of Apple_(technology_company), we can access the attributes of the company in a holistic way (e.g., products, corporate identity, etc.).

Although an entity has such inherent advantages over a term, it may suffer from errors in entity identification and disambiguation from free text due to the fact that the entity annotation is an automated process and therefore perfect accuracy can not be guaranteed. However, with the advance of entity recognition, such errors could be mitigated gradually and it is still a promising direction to explore how to leverage entities to improve retrieval performance.

In this paper, we propose to model the relevance using a *latent entity space*. Each dimension is represented by an entity, and a query is generated from a mixture of all the dimensions. Thus, we can factor the log-odds ratio in Equation (1) as follows:

$$
\begin{aligned}
p(\mathcal{R}=1|q,d) &\overset{\text{rank}}{=} \log \frac{p(q,d|\mathcal{R}=1)}{p(q|\mathcal{R}=0)p(d|\mathcal{R}=0)} \\
&\overset{\text{rank}}{=} \log \sum_{e\in\mathcal{E}} p(q,d|e,\mathcal{R}=1)p(e|\mathcal{R}=1) \\
&\overset{\text{rank}}{=} \log \sum_{e\in\mathcal{E}} p(q|d,e,\mathcal{R}=1)p(d|e,\mathcal{R}=1)p(e|\mathcal{R}=1) \\
&\overset{\text{rank}}{=} \sum_{e\in\mathcal{E}} p(q|d,e,\mathcal{R}=1)\cdot p(e|d,\mathcal{R}=1). \qquad (4)
\end{aligned}
$$

Similar assumptions are made as in Equation (3) during the derivation. As it is not practical to estimate the joint conditional probability $p(q|d,e,\mathcal{R}=1)$ directly, we use the linear interpolation of two individual conditional probabilities to estimate it by following previous work [51,5]:

$$
p(q|d,e,\mathcal{R}=1) = \lambda p(q|e,\mathcal{R}=1) + (1-\lambda)p(q|d,\mathcal{R}=1). \qquad (5)
$$

λ balances the importance of two probabilities. By plugging Equation (5) into Equation (4), we obtain:

$$
\begin{aligned}
p(\mathcal{R}=1|q,d) &\overset{\text{rank}}{=} \lambda \sum_{e\in\mathcal{E}} p(q|e,\mathcal{R}=1)\cdot p(e|d,\mathcal{R}=1) + (1-\lambda)p(q|d,\mathcal{R}=1)\sum_{e\in\mathcal{E}} p(e|d,\mathcal{R}=1) \\
&\overset{\text{rank}}{=} \lambda \sum_{e\in\mathcal{E}} \underbrace{p(q|e,\mathcal{R}=1)}_{\text{query projection}} \cdot \underbrace{p(e|d,\mathcal{R}=1)}_{\text{document projection}} + (1-\lambda)p(q|d,\mathcal{R}=1). \qquad (6)
\end{aligned}
$$

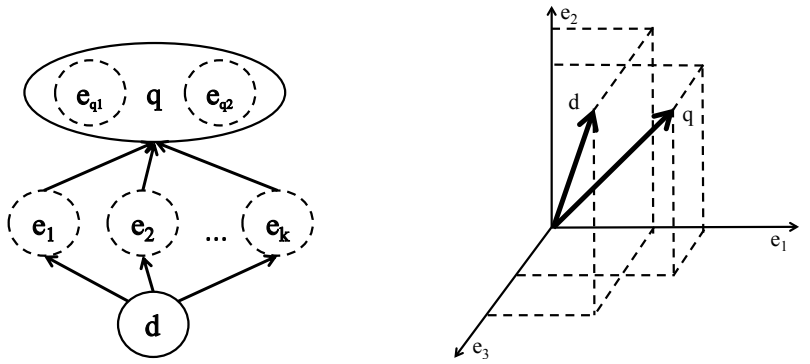
The first component essentially is LES. The underlying dependence network between all the variables involved in LES can be illustrated in Figure 3(a). For a given document d, we first choose an entity $e \in \mathcal{E}$ to represent one semantic aspect of d with probability $p(e|d,\mathcal{R}=1)$, and then generate the query q conditioned on e with probability $p(q|e,\mathcal{R}=1)$. The second component (i.e., $p(q|d,\mathcal{R}=1)$) is the query likelihood and can be estimated by existing language modeling based approaches (e.g., Equation (3)).

4.3 Estimation Details

We now discuss how to estimate the probability components of LES in Equation (6) in detail.

4.3.1 Document projection

$p(e|d,\mathcal{R}=1)$ can be interpreted as the projection of d on the dimension of e in the latent space, as illustrated in Figure 3(b). It can be estimated as the probability of e generated from θ_d (i.e., entity likelihood). Existing document retrieval models could be leveraged to estimate it, similar to the idea



(a) Dependence network of LES. (b) Projection in LES.

Fig. 3: Latent Entity Space.

of query likelihood. In this paper, we choose negative cross-entropy between entity model θ_e and document model θ_d, based on Kullback-Leibler (KL) divergence, one of the state-of-the-art retrieval models [55]:

$$p(e|d, \mathcal{R} = 1) = p(e|\theta_d, \mathcal{R} = 1) = \exp\Big(\sum_w p(w|\theta_e) \log p(w|\theta_d)\Big). \quad (7)$$

θ_e denotes the profile model of e. θ_d can be estimated through maximum likelihood estimation. To improve the estimation accuracy of document projection, we apply Dirichlet smoothing [54] to θ_d.

4.3.2 Query projection

$p(q|e, \mathcal{R} = 1)$ can be interpreted as the probability that q is generated from the profile model of e (i.e., θ_e). It actually serves as the weight of dimension represented by e in the latent space. We propose two methods to estimate the probability.

Unigram based approach. One straightforward way is to use the unigram LM [54]. By assuming terms in q are independent, the probability can be computed as:

$$p(q|e, \mathcal{R} = 1) = p(q|\theta_e, \mathcal{R} = 1) = \prod_{w \in q} p(w|\theta_e)^{n(w,q)}, \quad (8)$$

and $n(w, q)$ is the number of occurrences of term w in q.

Entity-similarity based approach. Since we have the entity annotations in the query, it would be interesting to study whether leveraging the query entities can deliver better performance. As query entities carry important aspects of information needs for a query, an important entity dimension should share high semantic similarity with them. Therefore, we propose to estimate the query projection based on the weighted sum of similarities between

e and each query entity $e_q \in E(q)$, where the weight is the importance of e_q. Formally, the probability can be estimated as:

$$p(q|e,\mathcal{R}=1) \propto \sum_{e_q \in E(q)} p(e_q|e,\mathcal{R}=1) \cdot p(e_q|m(e_q),q)$$
$$\propto \sum_{e_q \in E(q)} sim(\theta_{e_q},\theta_e) \cdot p(e_q|m(e_q),q). \quad (9)$$

θ_{e_q} denotes the profile model of e_q, $sim(\theta_{e_q},\theta_e)$ represents the similarity between θ_{e_q} and θ_e, and $p(e_q|m(e_q),q)$ is the posterior probability provided in the annotation data, as described in Section 3. Since both θ_{e_q} and θ_e are of the same type, any pairwise symmetric distance-based information similarity measure can be adopted to estimate $sim(\theta_{e_q},\theta_e)$. In this paper, we choose cosine similarity, and leave other measures as future work.

Since the unigram-based approach makes the term independence assumption and does not use any information about the entity annotations in the query, it would not capture the semantic correlation between q and e as well as the entity-similarity based method. We expect the entity-similarity based method to work better than the unigram-based approach, which is confirmed by the experimental results in Section 5.4.2.

4.4 Estimation of Entity Profile

The estimations of both $p(e|d,\mathcal{R}=1)$ and $p(q|e,\mathcal{R}=1)$ require θ_e, i.e., the entity profile model, which represents the characteristics of e. Since the relevance between d and q is estimated through θ_e in LES, a comprehensive and accurate estimation of θ_e clearly is crucial to the performance. We propose two methods to estimate θ_e from the document collection $\mathcal{D}$ and knowledge base $\mathcal{K}$ as follows.

4.4.1 Build entity profiles from scratch

One entity may be mentioned in multiple documents, and each document carries some information about the entity. Although a single document could only provide partial information about the entity in certain aspects, it is possible to construct a complete picture of the entity by aggregating information from all the documents mentioning the entity, similar to the process of solving a jigsaw puzzle. Specifically, we adopt language modeling to estimate θ_e as follows:

$$p(w|\theta_e) = \frac{1}{|\mathcal{C}(e)|} \sum_{c(e) \in \mathcal{C}(e)} p(w|c(e)),$$

where $c(e)$ is a context of e from a document and $\mathcal{C}(e)$ is the set of all contexts in which e occurs. Basically $c(e)$ includes a sequence of σ terms before and after $m(e)$, and $pos(e)$ is right at the center of $c(e)$. The underlying assumption is

that terms around an entity mention carry pieces of jigsaw-like entity-related information, including attributes, relations with other entities, etc. We now discuss how to estimate $p(w|c(e))$.

A straightforward solution is to use maximum likelihood estimation:

$$p(w|c(e)) = \frac{n(w, c(e))}{\sum_{w'} n(w', c(e))}, \tag{10}$$

where $n(w, c(e))$ is the number of occurrences of w in $c(e)$. Although the bag-of-words assumption works empirically well in language modeling based retrieval, it does not always hold in the estimation of entity profile model as terms closer to entity are more relevant to the entity than terms farther away. Therefore, it is necessarily important to incorporate proximity information into the estimation of entity profile model.

An alternative way is to use a proximity-based approach to model the representation for entity-bearing documents. Motivated by the previous study [42], we can estimate $p(w|c(e))$ as follows:

$$p(w|c(e)) = \frac{1}{Z} \sum_{i=1}^{|c(e)|} \delta_{c(e)}(i, w) k(w, c(e)), Z = \sum_{i=1}^{|c(e)|} k(w, c(e)), \tag{11}$$

where Z is a normalization constant to make sure $p(w|c(e))$ follows a probability distribution, $\delta_{c(e)}$ is an indicator function:

$$\delta_{c(e)}(i, w) = \begin{cases} 1 \text{ if term at position } i \text{ in } c(e) \text{ is } w \\ 0 \text{ otherwise.} \end{cases}$$

Different from maximum likelihood estimation, a proximity-based coefficient is imposed on each term $w \in c(e)$ so that terms closer to e would receive more weight than others. The kernel function $k(w, c(e))$ actually enables the incorporation of proximity information. Any non-uniform, non-increasing function can serve as proximity functions. One commonly used kernel function is the Gaussian kernel:

$$k(w, c(e)) = \mathcal{N}(w, c(e), \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big[\frac{-(pos(w) - pos(e))^2}{2\sigma^2}\Big],$$

where $pos(w)$ is the position of w in $c(e)$ and $pos(e)$ is the position of $m(e)$ in $c(e)$, respectively. In this paper, we use the Gaussian kernel, and leave other kernel functions as future work. We fix σ to 40 based on preliminary results. Experimental results in Section 5.4.3 confirm that proximity information helps on estimation of entity profile.

4.4.2 Leverage existing knowledge bases

Compared to Web documents, in which entity-related information are scattered, knowledge bases provide a portal to access full spectrum of information about entities in a much easier way. Since manual efforts are involved in the curation of knowledge bases, high quality information is guaranteed. In a knowledge base, an entity is represented as a structured document with multiple fields, each field is associated with some type of semantic aspect. An intuitive approach is to merge all (or some) fields as one document and apply maximum likelihood estimation over it:

$$p(w|\theta_e) = \frac{n(w, kb(e))}{\sum_{w'} n(w', kb(e))}. \tag{12}$$

In the problem setup of this paper, we use Freebase [8] as the knowledge base, and choose the description field (i.e., `/common/topic/description`) as the document to represent the profile of entity, as it provides much richer textual information than other fields. In most cases, the description field is fetched from the introduction section of the corresponding Wikipedia entry automatically, and is complemented by Freebase community editors manually for the entities without the corresponding Wikipedia entries. In general, the description field provides a piece of concise text describing the entity.

4.5 Learning to Balance LES and Query Likelihood

As shown in Equation (6), the relevance score of a document is a linear combination of LES and query likelihood estimation, and the interpolation coefficient λ controls the importance of these two components. Intuitively, the value of λ should relates to the characteristics of a query. For example, if a query is not about any entities, λ would need to have a very small value. On the contrary, if entities play the most important role in a query, we would need to set the value of λ to a larger value. We propose to learn the value of λ for each query based on the following two features.

- **Entity coverage**. This feature, denoted as $cov(q)$, measures how much information about a query is covered by entity terms. In particular, we compute the ratio of terms which are mapped to entities according to the entity annotations:

$$cov(q) = \frac{\sum_{e \in E(q)} n(e)}{n(q)},$$

  where $n(e)$ and $n(q)$ represent the number of terms in e and q respectively. When the coverage of a query is low, the query does not contain much information about the entities and the value of λ would be smaller.

– **Entity novelty**. This feature, denoted as $nov(q)$, measures how much novel information that the entity profile can bring given the query. A natural way of measuring such novelty is to use the KL-divergence between the the relevance model of the query [34] and the entity model:

$$nov(q) = \sum_{e \in \mathcal{E}} D_{KL}(\theta_q^R || \theta_e),$$

where θ_q^R is the relevance model of q, estimated by the top retrieved documents by query likelihood (i.i.d. sampling). When the entity novelty is high, we would give more weight to LES, i.e., setting λ to a larger value, since it could bring additional relevant information.

We choose Support Vector Machine (SVM) [17] regression[3] with the Gaussian kernel to estimate λ. More specifically, we apply n-fold cross-validation to train the model on n-1 fold labeled queries with optimal value of λ and test on the remaining one fold. More details will be discussed in Section 5.2.

4.6 Implementation Details

Score Normalization: Since the probabilities of LES and query likelihood in Equation (6) are actually estimated by retrieval scores, they may not be on the same scale, and it is necessary to apply normalization before interpolation. Since we are not aware of the mean and deviation of probabilities for each query, we transform the probabilities to the ranking of documents:

$$S(q,d) = \lambda M(R_{les}(q,d)) + (1-\lambda) M(R_{ql}(q,d)), \tag{13}$$

where $R_{les}(q,d)$ and $R_{ql}(q,d)$ are the rankings of d with regard to q by scores of LES and query likelihood respectively. $M(R(q,d)) = (\max_{d'} R(q,d') - R(q,d))/\max_{d'} R(q,d')$ maps the ranking to a linear scale score in $[0,1)$.

Reduced Entity Space: When coming to the implementation of LES, it is crucial to decide which entities should be selected to serve as dimensions. Theoretically, the entity space should include all the entities in $\mathcal{E}$. However, it would be computationally prohibitive, and more importantly, due to the nature that we can not get the exact profile of entity, the more entities selected, the more likely that LES would be "distorted" by the inaccurate estimation of θ_e and thus worse performance. On the other hand, if only few entities are selected, it is possible that some important aspects would be missed. To balance the tradeoff, we choose a set of most relevant k entities, selected based on the query projection $p(q|e, \mathcal{R} = 1)$ as shown in Section 4.3.2, to approximate $\mathcal{E}$.

[3] We also tried other linear regression methods, and they could not deliver better performance.

Note that the estimation of entity profile from collection can be done offline. To further reduce the computational cost, we only apply LES to re-rank the top-n documents ranked by query likelihood $p(q|d, \mathcal{R} = 1)$. The choices of k and n will be explored in Section 5.5.

5 Experiments

All the queries and run results in this paper are available online[4].

5.1 Experimental Setup

We choose ClueWeb09 Category B, a standard TREC dataset to conduct experiments, as it is a representative large-scale English Web collection used in many tracks of TREC recently. Freebase [8] is selected to serve as the accompanying knowledge base, as it provides adequate coverage on the entities in the Web. To link ClueWeb09 with Freebase, we leverage Freebase Annotations of ClueWeb Corpora, v1 (FACC1) [30], a dataset built by Google which provides entity extraction and linking to Freebase entries for documents in ClueWeb09. About 70% of documents in ClueWeb09 collection have valid annotations. Queries are taken from TREC Web Track 2009 to 2012 [10–13]. In particular, Google provides *automatic* entity annotation for 94 of all the 200 queries[5] in the *description* field, making it feasible to evaluate the performance of LES on entity-bearing queries systematically. Waterloo Spam Rankings [16] is employed to filter out spam documents (percentile-score threshold is set to 70 based on recommendation). Porter stemmer is applied, and stop words are removed for both entity profile estimation and document retrieval.

According to the explanation from Google, the context of one entity consists of both local (terms around the entity mention) and global (entities that occur throughout the document as context feature) information. The posterior probabilities are estimated based on the learning using a combination of labeled and unlabeled data.

We design a set of experiments to investigate the following research questions:

1. Can LES capture semantic relevance for entity-bearing queries? (Section 5.2)
2. Is the semantic relevance feature captured by LES complementary to the state-of-the-art LM approaches? (Section 5.3)
3. Is LES a robust approach compared with the state-of-the-art LM approaches? (Section 5.4)

To evaluate the performance, we choose two measures used in TREC Web track as primary measures: (1) nDCG@20 (normalized Discounted Cumulative Gain at rank 20), (2) ERR@20 (Expected Reciprocal Rank at rank 20).

[4] http://xtliu.com/data/les/

[5] http://lemurproject.org/clueweb09/related-data.php

Besides, as numerous studies suggest that Web search users mostly focus on the top 10 results in the first search result page, we also report the both measures with cutoff at rank 10 (i.e., nDCG@10 and ERR@10) as complementary measures.

We compare the proposed LES methods with the following five baselines:

- **DIR**: Dirichlet prior smoothing retrieval method [54], one of the state-of-the-art keyword-based retrieval methods;
- **RM3**: Relevance Model [34], one of the state-of-the-art feedback methods;
- **LCE**: Latent Concept Expansion [40], a generalization of relevance models with term dependence;
- **LDA**: Latent Dirichlet Allocation based document modeling [51];
- **KC**: Key concept based approach for verbose queries [5][6].

Note that both RM3 and LCE represent the state-of-the-art query expansion methods. The implementations for RM3 and LCE are provided by Ivory[7]. LDA and KC represent the concept based document and query modeling approaches respectively.

5.2 Effectiveness of LES

We conduct experiments to evaluate the proposed LES methods. When implementing the LES methods, we use the entity-similarity based approach as described in Equation (9) to estimate the query projection, employ proximity based approach (Equation 11) to estimate entity profile on document collection (denoted as **LES-COL**) and maximum likelihood approach (Equation 12) to estimate entity profile on Freebase (denoted as **LES-FB**). The results of LES methods are generated by re-ranking of top 90 ranked documents of DIR.

We conduct experiments using five-fold cross-validation for all the baselines as well as LES methods. Specifically, queries are randomly divided into five subsets, each subset is used as the test set in turn, while other queries serve as labeled training set for parameter learning through extensively searching over the entire parameter space. Testing results from all five subsets are then aggregated and average scores over all the 94 queries are reported in Table 1.

We observe that both LES-COL and LES-FB outperform all baselines, and the improvements of LES-COL over all baselines are statistically significant, demonstrating the effectiveness of LES. It is also interesting to note that most of the state-of-the-art methods are unable to significantly improve the performance over the DIR baseline, indicating the need of more effective ranking strategies for entity-bearing queries.

In particular, we notice that LES (according to Equation (6)) shares some similarity with the key concept (KC) approach proposed by Bendersky and Croft [5], however, they differ in two aspects: (1) LES leverages entity profile to estimate the probabilities, while the key concept approach only uses entity

[6] We use the Freebase entity annotations directly as weighted key concepts in the query.

[7] http://lintool.github.io/Ivory/

Table 1: Results of five-fold cross-validation.

Models	nDCG@20	ERR@20	nDCG@10	ERR@10
DIR	0.2316	0.1386	0.2404	0.1320
RM3	0.2460	0.1463	0.2513	0.1401
LCE	0.2765	0.1556	0.2800	0.1469
LDA	0.2652	0.1439	0.2657	0.1357
KC	0.2790^{D}	0.1523^{DR}	0.2837^{D}	0.1426^{D}
LES-COL	$\mathbf{0.3059}^{DRLAK}$	$\mathbf{0.1829}^{DRLA}$	$\mathbf{0.3064}^{DRLA}$	$\mathbf{0.1751}^{DRLA}$
LES-FB	0.2862^{DR}	0.1732	0.2897^{DRL}	0.1660

^{D}, ^{R}, ^{L}, ^{A} and ^{K} denote improvements over DIR, RM3, LCE, LDA and KC are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

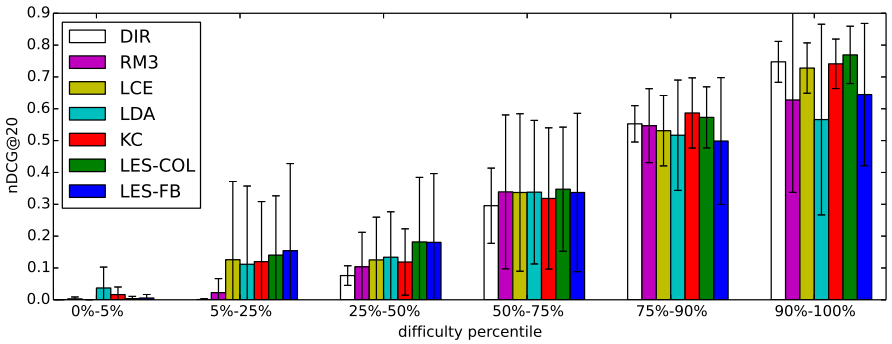


Fig. 4: Mean performance (nDCG@20) of different query difficulties. Queries are grouped based on the percentile of DIR. Error bars represent standard deviation.

names. (2) LES can select entities that are not from the query, while the key concept approach is limited to entities within the query. The improvements of LES-COL and LES-FB over KC implies that entity profile is better at capturing semantic relevance than entity names as concepts. Besides, related entities not in query also contribute to the improvement of retrieval performance. This is further confirmed in Section 5.4.4.

To further investigate the performance of LES on queries with different difficulty levels, we group all the queries into 6 sets based on the percentile of DIR baseline, and plot the average performance (nDCG@20) in each set in Figure 4. The hardest 5% queries are grouped in the left-most column, while the easiest 10% queries are grouped in the right-most column. We observe that both LES-COL and LES-FB could improve more on hard queries over DIR before 50% percentile than other baselines. When the query gets easier, the improvements become smaller. It is interesting to note that even for very easy queries (above 90% percentile), LES-COL could still outperform DIR while

other baselines perform worse than DIR. In summary, LES-COL outperforms DIR across all query difficult spectrum, demonstrating its strong effectiveness and robustness.

We did some analyses about the hard and easy queries and found that on average for hard queries (below 25% percentile) the entity novelty score of related entities (as described in Section 4.5) are 13.26% higher than easy queries (above 25% percentile). It suggests that for easy queries the information need is already clearly represented in query and related entities could not contribute much, while for hard queries related entities could bring more complementary information and thus LES has more potential to improve performance.

We now use an example query to explain how the proposed LES methods can improve the search accuracy. Consider query #11 "I'm looking for information to help me prepare for the GMAT exam", which is improved by LES-COL from 0.1429 to 0.3434 in terms of nDCG@20. The top three entities in LES are "GMAT", "Graduate Management Admission Council (GMAC)" and "The Princeton Review". "GMAT (Graduate Management Admission Test)" is the annotated query entity, which reflects the most important aspect of the information need. "Graduate Management Admission Council (GMAC)" is the administrator to GMAT, and "The Princeton Review" is an American-based standardized test preparation and admissions consulting company which provides GMAT Test Preparation service. Clearly, these two entities are related to the query and can provide complementary aspects about the information need.

We also analyze some queries on which LES fails. For example, query #179 "find a timeline for African American in the United States" is hurt by both LES-COL and LES-FB and the performance drops from 0.1902 to 0 in terms of nDCG@20. The top 5 entities in LES are "African American", "United States", "Southern United States", "Chinese American" and "White American", the first two of which are query entities. Among the three related entities, "Chinese American" and "White American" are similar to "African American" in terms of category, and "Southern United States" is part of "United States", but they are not directly related to the query, thus LES diverges from the original information need and fails to retrieve relevant documents. The failure of LES is mainly due to the ignorance of non-entity term "timeline", which implies the history aspect is desired by the query. Inspections on the relevant documents suggest that related entities like "Colonial History of the United States", "American Revolutionary War", "American Civil War" would help. We expect that the performance of LES could be improved if we incorporate the entity relations feature from knowledge bases into the selection of related entities, and leave it as future work.

The comparison between LES-COL and LES-FB reveals that entity profiles estimated from the document collection are more effective than those from Freebase. Our analyses suggest three reasons: (1) The quality of automatic query entity annotation is not very good, as some entities are labelled incorrectly and some could not be annotated. (2) The coverage of Freebase entity profile is not complete, especially for tail entities. (3) The Freebase entity

Table 2: Comparison with Relevance Model and Latent Concept Expansion.

Models	nDCG@20	ERR@20	nDCG@10	ERR@10
RM3	0.2460	0.1463	0.2513	0.1401
LES-COL	$\mathbf{0.2884}^{R}$	$\mathbf{0.1814}^{R}$	**0.3042**	$\mathbf{0.1749}^{R}$
LES-FB	0.2823	0.1743	0.2844	0.1658
LCE	0.2765	0.1556	0.2800	0.1469
LES-COL	$\mathbf{0.3065}^{RL}$	$\mathbf{0.1908}^{RL}$	$\mathbf{0.3310}^{RL}$	$\mathbf{0.1854}^{RL}$
LES-FB	0.2851^{R}	0.1764^{R}	0.2837	0.1686

Results are under five-fold cross-validation settings. ^{R} and ^{L} denote improvements over RM3 and LCE are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

profile does not reflect the exact statistics of terms in the document collection. It suggests that only with entity annotations on document collection, we can already reach good performance. Besides, we try to combine the entity profiles from document collection and Freebase, and it could only bring marginal improvements. We leave this as our future work.

5.3 Complementarity of LES

Since LES is capable of capturing entity-based semantic relevance, which is an important feature on relevance, we hypothesize that this feature is complementary to the existing term-space based approaches. To verify our hypothesis, we choose RM3 and LCE, two state-of-the-art LM based approaches as baselines for query likelihood estimation in Equation (6). The results of LES based approach are based on the re-ranking of top 90 ranked documents of RM3 and LCE accordingly.

Table 2 summarizes the results under five-fold cross-validation settings. Interestingly, we find that after interpolation with LES based approaches, the results can be improved significantly for both RM3 and LCE. This verifies our hypothesis that the semantic relevance feature captured by LES is complementary to term-space based approaches. Besides, LES-COL performs better than LES-FB, which is consistent with the observation as in Table 1.

5.4 Extensive Analyses

5.4.1 Effectiveness of learning $\boldsymbol{\lambda}$ for each query

As discussed in Section 4.5, the interpolation coefficient λ in Equation (6) is an important factor to the performance. We now examine the effectiveness of our approach on learning λ. We conduct two sets of experiments: (1) tuning the parameter with five-fold cross-validation, but without the learning of λ (i.e., λ

Table 3: Comparison of results on learning $\boldsymbol{\lambda}$.

	no-learning		**learning**	
Method	nDCG@20	ERR@20	nDCG@20	ERR@20
DIR	0.2316	0.1386	0.2316	0.1386
LES-COL	0.2905	0.1653	$\mathbf{0.3059}^D$	$\mathbf{0.1829}^D$
LES-FB	0.2633	0.1617	$\mathbf{0.2862}^D$	**0.1732**

Results are under five-fold cross-validation settings. ^{D} denotes improvements over DIR are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

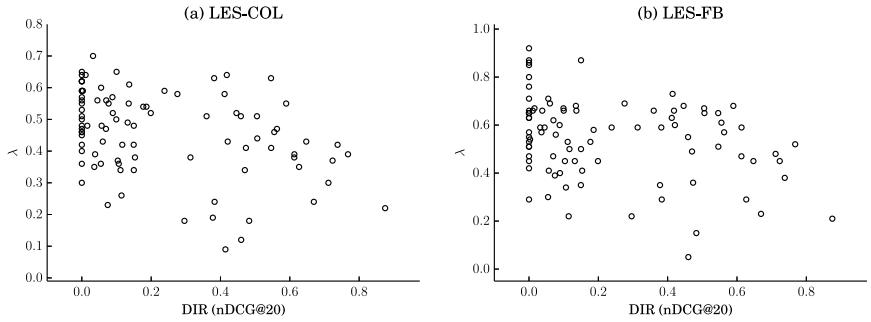


Fig. 5: Correlation between query difficulty (nDCG@20 of DIR) and learned λ.

is set the same for all queries), denoted as **no-learning**. (2) tuning the parameter with five-fold cross-validation with the learning of λ, which is essentially the same as in Section 5.2 and denoted as **learning**. All the LES based results are based on the re-ranking of top 90 documents from DIR. Results are summarized in Table 3. By comparing the same LES based approaches in **no-learning** and **learning**, we observe that incorporating the learning of λ could improve effectiveness significantly.

To further investigate the effectiveness of learned λ, we plot the distribution of queries by difficulty and learned λ for both LES-COL and LES-FB, as shown in Figure 5. The x-axis represents the difficulty of query, measured by nDCG@20 of DIR, and y-axis represents the prediction of λ. Clearly, we could observe that there is linear correlation between them, demonstrating that our learning approach could predict λ based on query difficulty levels appropriately. For difficult queries λ would be set to high to raise the impact of LES on the final document ranking, while for easy queries on which query likelihood could perform well, λ would be set to low to given more weight to query likelihood.

Table 4: Comparison on query projection.

	unigram		**sim**	
Models	nDCG@20	ERR@20	nDCG@20	ERR@20
DIR	0.2316	0.1386	0.2316	0.1386
LES-COL	0.2738	0.1550	**0.3059**^{D}	**0.1829**^{D}
LES-FB	0.2491	0.1529	**0.2862**^{D}	**0.1732**

Results are under five-fold cross-validation settings. ^{D} denotes improvements over DIR are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

Table 5: Comparison on different kernel functions.

Models	nDCG@20	ERR@20	nDCG@10	ERR@10
DIR	0.2316	0.1386	0.2404	0.1320
constant	0.2690	0.1535	0.2761	0.1466
Gaussian	**0.3059**^{D}	**0.1829**^{D}	**0.3064**^{D}	**0.1751**^{D}

Results are under five-fold cross-validation settings. ^{D} denotes improvements over DIR are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

5.4.2 Query projection estimation

We have discussed two possible ways of estimating query projection, and we now compare their effectiveness empirically. The first method (**unigram**) is unigram-based approach as shown in Equation (8), and the second (**sim**) is the entity-similarity based approach as shown in Equation (9). DIR is chosen to estimate the query likelihood score. Results are reported in Table 4.

Obviously, entity-similarity based approach outperforms language modeling based approach in all settings, implying that entity annotations in query do help on finding important related entities for LES.

5.4.3 Kernel function

Recall that when entity profile is estimated from the document collection, we incorporate proximity into the estimation (in Equation 11). To understand the effectiveness of proximity, we compare the default Gaussian kernel function we choose (denoted as **Gaussian**) with constant kernel function $k(w, c(e)) = 1$ (which is equivalent with Equation 10) where no proximity information is incorporated (denoted as **constant**). Results on LES-COL are presented in Table 5. Clearly, Gaussian kernel outperforms constant kernel, confirming that proximity contributes to the estimation of entity profile.

Table 6: Results using query entities only for LES.

Models	nDCG@20	ERR@20	nDCG@10	ERR@10
DIR	0.2316	0.1386	0.2404	0.1320
LES-COL-QENT	**0.2767**^{D}	0.1491	**0.2771**	0.1414
LES-FB-QENT	0.2660	**0.1506**	0.2745	**0.1435**

Results are under five-fold cross-validation settings. ^{D} denotes improvements over DIR are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

5.4.4 Query entities only for LES

We notice that query entities are ranked at top by query projection score, as query entities have high similarity to themselves than others according to the similarity measure we adopted in Equation (9). This is reasonable as the query entity themselves reflect important aspects of the information need. It is interesting to explore whether the query entities are enough for LES. We design a set of experiments by only using query entities to construct LES. The query projection scores are estimated by Equation (9) as well. We denote this method as LES-COL-QENT and LES-FB-QENT for entity profile estimated from document collection and Freebase respectively. Results are shown in Table 6. By comparing the results in Table 4, we find that LES based on query entities only can improve the performance. However, the performance is inferior to LES-COL and LES-FB which use more than query entities, implying that related entities can provide additional aspects complementary to query entities.

5.4.5 Robustness

To investigate the robustness of our models in a quantitative approach, we report the numbers of queries which are improved/hurt (and by how much) compared with DIR under five-fold cross-validation setting. A robust method is expected to improve the performance for more queries over the baseline and hurt less [50]. The results under five-fold cross-validation (same as reported in Table 1) are illustrated in Figure 6. The x-axis represents the relative improvements/degradations in nDCG@20, clustered in several groups. The y-axis represents the number of queries in each group. The bars to the left of $(0\%, 25\%]$ represent queries on which other models perform worse than the DIR baseline, and the other bars on the right size represents queries on which other models perform better.

Obviously, both LES-COL and LES-FB exhibit stronger robustness than all other baselines. In particular, LES-COL improves 46 queries and hurts 28, LES-FB improves 41 queries and hurt 33, whereas RM3 improves 39 queries and hurts 31, LCE improves 38 queries and hurts 36, LDA improves 43 queries and hurt 31, CPT improves 37 queries and hurts 26, respectively.

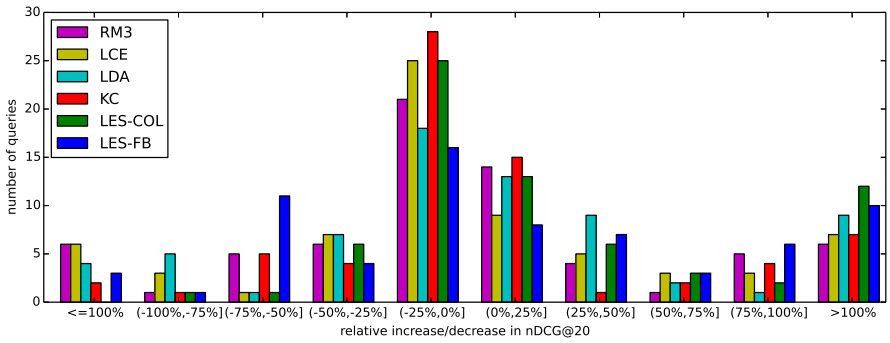


Fig. 6: Histogram of queries when applied with different models compared with DIR.

5.5 Parameter Sensitivity

We now report the performance sensitivity of parameters used in our methods.

The first parameter is k, which is the number of dimensions in LES as mentioned in Section 4.6. As shown in Figure 7(a), the performance increases with k when k is less than 3, and best performance is reached when $k = 3$ for both LES-COL and LES-FB. After k passes 3, the performance of LES-FB begins to drop gradually, which LES-COL remains relatively stable. The main reason is that the Freebase entry are manually edited, thus the term statistics of Freebase entity model are different from document collection. The more entities are selected in LES, the more likely the estimation of document projection gets distorted. On the contrary, entity models estimated from the document collection are sampled over multiple documents, therefore smoother and more robust with regard to the number of dimensions. It suggests that LES-COL is a better choice than LES-FB in terms of robustness.

The second parameter is n, which represents the number of documents LES re-ranks over the results of language modeling approach. As shown in Figure 7(b), we find that both LES-COL and LES-FB exhibits the similar performance trend on n. When n is small, the potential of LES is limited as only a few documents get involved and the ranking of documents below n will remain unchanged. As n increases, more documents previously mis-penalized by language modeling approach will get the chance to be promoted by LES, leading to the performance increase. After n is greater than 100, the performance remains stable. As the computational cost of LES is proportional to n, it is a suggested to be set around 100 to reach both good effectiveness and efficiency.

The third parameter is λ, which controls the influence of interpolation between LES and query likelihood in Equation (13). We set the same λ for all queries with different values, and report the results in Figure 7(c). We observe that the performance increases with λ, as LES introduces more improvements.

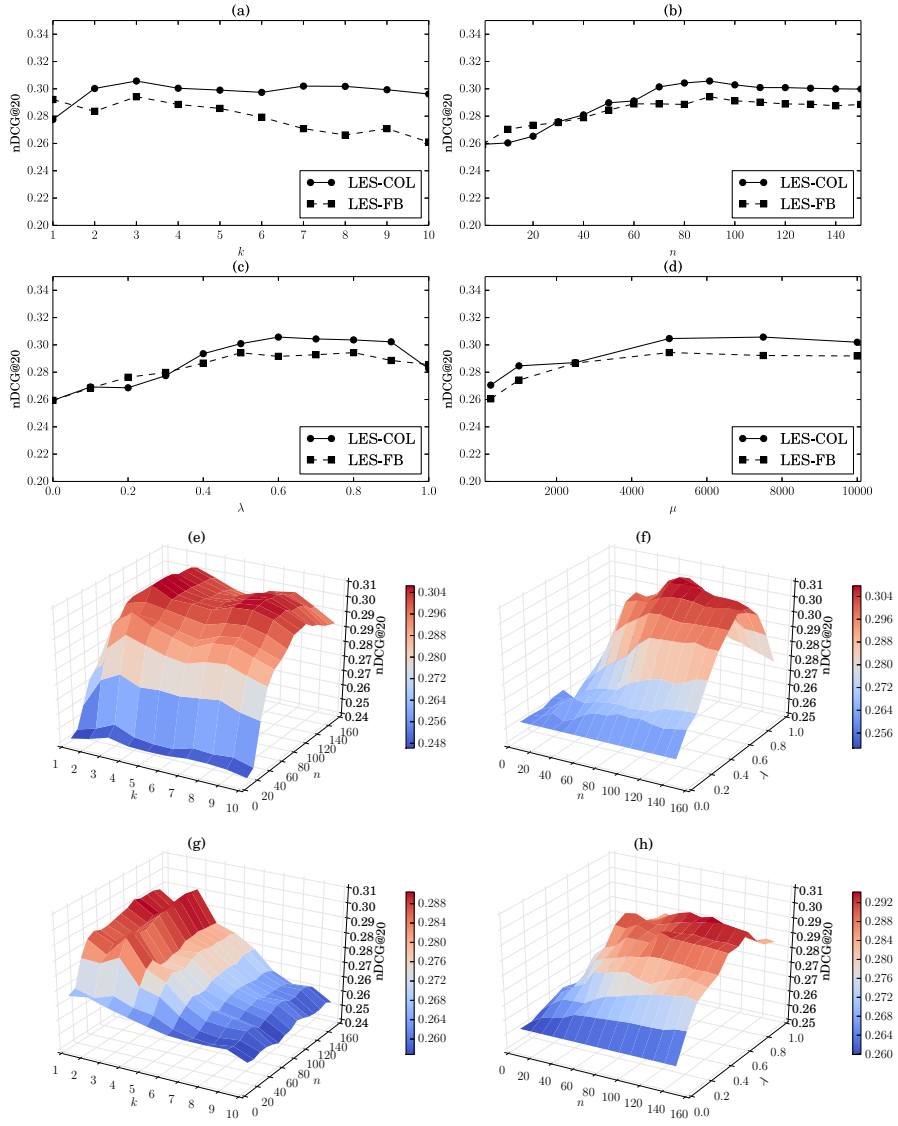


Fig. 7: Parameter sensitivity.

Optimized performance is reached when $\lambda = 0.6$. After that, the performance starts to decrease slowly. When no training data is available, λ is suggested to be set between 0.5 and 0.7.

The fourth parameter is μ, the Dirichlet smoothing parameter for document model θ_d in the estimation of document projection (Equation 7). As shown in Figure 7(d), we observe that when μ is less than 5,000, the performance

increases gradually with μ. The optimal performance is reached when μ is around 5,000. After that, the performance remains stable with a little loss. The observations are similar to previous study [54] on language modeling approach.

Furthermore, we also plot the joint distribution between k, n and λ to better understand the correlation between them. Figure 7(e) demonstrates the joint distribution of k and n, and figure 7(f) shows the joint distribution of n and λ, for LES-COL respectively. We observe that the optimal performance is reached when $k \in [2, 4]$, $n \in [80, 100]$, $\lambda \in [0.5, 0.7]$.

Figure 7(g) and figure 7(h) illustrate the joint distribution of k and n, n and λ on LES-FB respectively. It is interesting to note that LES-FB is more sensitive on k and the performance drops fast as k increases, which can also be observed on Figure 7(a). This is mainly due to the fact that the entity profile for LES-FB are estimated from Freebase, and the entity model deviates from the collection model for low-ranked entities and therefore would hurt the performance. While for LES-COL, as the entity model is estimated from the document collection, it is much "smoother" even for low-ranked entities and would not hurt the performance much. The other observations are similar as on LES-COL. The optimal performance is reached when $k \in [1, 3]$, $n \in [90, 120]$, $\lambda \in [0.6, 0.8]$.

6 Extensive Experiments on TREC data (2009-2014)

6.1 Comparison with Entity Query Feature Expansion

We are aware that there has been a few work done on leveraging the FACC1 [30] data to improve Web retrieval performance. In particular, Dalton et al. [20] proposed Entity Query Feature Expansion (EQFE) model which integrates various entity related features (e.g., related entities, categories, Wikipedia, entity context, collection feedback, etc.) to model the query-document relevance based on a general learning-to-rank framework. They evaluated their model on the TREC Web track data from 2009 to 2012, and provided an extended query entity annotation list based on manual revision. The revised query list consists of 191 out of 200 queries with valid Freebase entity annotations, with 97 more queries than the 94 queries along with the FACC1 data set. Besides, the entity annotations for many queries have been fixed manually [20], thus are much better in terms of both quantity and quality. It is therefore interesting to conduct experimental evaluation with the extended query list to see whether our model could benefit from the improved query annotation.

To conduct a fair comparison with the Entity Query Feature Expansion model, we follow the exact description by Dalton et al. [20] to use ClueWeb09 Category B as data collection, and employ Waterloo Spam Rankings [16] to filter out spam documents (percentile-score threshold is set to 60 according to their description). Besides, stop words are removed based on the INQUERY 418 world stop list with web-specific terms including "com", "html", "www", etc.

For all the 200 queries from 2009 to 2012 Web track data, we use the *title* field to perform retrieval based on Dirichlet prior smoothing retrieval method as baseline (denoted as **DIR**), in contrast to our previous experiments in Section 5 where *description* field is used for retrieval. The extended 191 query annotation list is employed for our LES based models. All the 200 queries are used for evaluation, for the remaining 9 queries without annotation, we fall back to use the results of DIR instead. Similarly, we use the entity-similarity based approach as described in Equation (9) to estimate the query projection, and employ proximity based approach in Equation (11) to estimate entity profile on document collection (denoted as **LES-COL**) and maximum likelihood approach in Equation (12) to estimate entity profile on Freebase (denoted as **LES-FB**). Results of LES-COL and LES-FB are based on the re-ranking of top ranked documents of DIR. All the result evaluations are reported in nDCG@20 and ERR@20, as well as nDCG@10 and ERR@10, similar as in Section 5.

We directly retrieve the final run files (for all 198 valid queries[8]) provided online by Dalton et al.[9] and use the TREC standard evaluation script and judgment to evaluate them to make sure the comparison is fair. We use all of their four reported runs as baselines:

- **SDM**: Sequential Dependence Model [39].
- **WikiRM1**: External feedback model which uses Wikipedia as text collection and extracts terms from top ranked documents.
- **SDM-RM3**: SDM extended with collection relevance model [34] as feedback.
- **EQFE**: Entity Query Feature Expansion model which leverages 42 features from collection and Freebase knowledge base.

According to their description, the top three baselines are state-of-the-art word-based retrieval and expansion models. Parameters for all the baselines and our runs are tuned based on five-fold cross-validation, and results are summarized in Table 7.

We observe that both LES-COL and LES-FB demonstrate superior performance over all the five baselines, confirming the effectiveness of LES. Compared to the results in Table 1, it is interesting to note that LES-FB now performs better than LES-COL, implying that LES-FB could further benefit from the improvement of entity annotation quality in queries.

We notice that EQFE could not outperform all the top three baselines, and the observations are consistent with those reported before [20]. Dalton et al. argue that 37.4% relevant documents in ClueWeb09 Category B do not contain one explicit query entity (and the largest portion are Wikipedia documents), and 57% of relevant documents with at least one entity annotation do not contain explicit query entity, causing their proposed EQFE model failed to retrieve such documents. The fact that their model heavily relies on the valid

[8] Actually they do not contain results for query #95 and #100, as there are no official judgments from TREC. Therefore, it is essentially the same as 200 queries in our case.

[9] http://ciir.cs.umass.edu/downloads/eqfe/

Table 7: Comparison between four baselines and LES based models (200 queries, title field only).

Models	nDCG@20	ERR@20	nDCG@10	ERR@10
SDM	0.2140	0.1363	0.2165	0.1275
WikiRM1	0.2256	0.1529	0.2328	0.1447
SDM-RM3	0.2204	0.1478	0.2311	0.1397
EQFE	0.2116	0.1400	0.2192	0.1322
DIR	0.1992	0.1317	0.2006	0.1227
LES-COL	0.2409^{SED}	0.1716^{SWED}	0.2539^{SWED}	0.1637^{SWED}
LES-FB	$\mathbf{0.2560}^{SWRED}$	$\mathbf{0.1990}^{SWRED}$	$\mathbf{0.2728}^{SWRED}$	$\mathbf{0.1917}^{SWRED}$

^{S}, ^{W}, ^{R}, ^{E} and ^{D} denote improvements over SDM, WikiRM1, SDM-RM3, EQFE and DIR are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

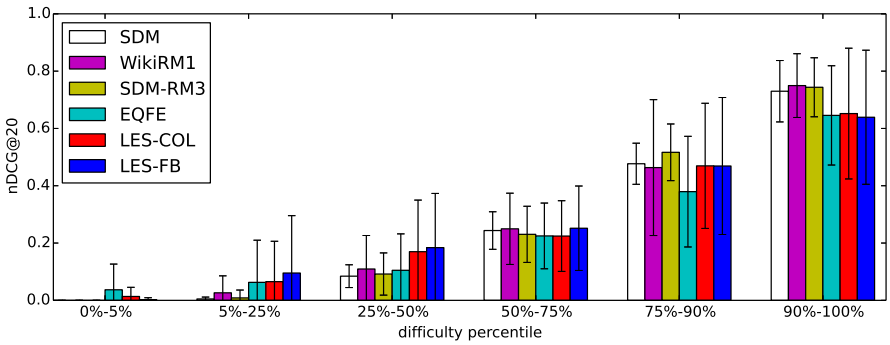


Fig. 8: Mean performance (nDCG@20) of different query difficulties on ClueWeb09 Category B data (TREC 2009 - 2012). Queries are grouped based on the percentile of SDM. Error bars represent standard deviation.

entity annotation in the relevant documents as well as the mention of query entities makes it vulnerable to suffer from the low quality (either recall or precision) of entity annotations on relevant documents. In contrast, LES does not directly use the entity annotations in relevant documents to model the relevance, as the entity models are estimated on the whole document collection (for LES-COL) or from Freebase (for LES-FB), therefore are more robust with regard to low entity annotation quality for partial documents.

To further compare the performance on queries with different difficulty levels, we group all the queries into 6 sets based on the percentile of SDM baseline, and plot the average performance (nDCG@20) in each set in Figure 8. Note that Dalton et al. claim that the strength of EQFE is the ability to improve hard queries (below 50% percentile). We observe that LES-COL and LES-FB could also improve hard queries, and the improvements are larger that EQFE. In fact, both LES-COL and LES-FB outperform EQFE across almost

Table 8: Results of five-fold cross-validation on TREC 2013 Data.

Models	nDCG@20	ERR@20	nDCG@10	ERR@10
DIR	0.2141	0.1239	0.1988	0.1138
LCE	0.2171	0.1347	0.2091	0.1235
KC	0.2183	0.1248	0.2048	0.1231
LES-COL	0.2288^{D}	0.1410	0.2235^{D}	0.1322
LES-FB	$\mathbf{0.2583}^{DLK}$	$\mathbf{0.1642}^{DLK}$	$\mathbf{0.2467}^{DL}$	$\mathbf{0.1550}^{DL}$

^{D}, ^{L}, and ^{K} denote improvements over DIR, LCE and KC are statistically significant at 0.05 level based on Wilcoxon signed-rank test, respectively.

the full query difficulty spectrum (i.e., [5%, 100%]). The only exception range is [0%, 5%), and we think the advantage of EQFE on extremely hard queries are based on the integration of 42 different features in different levels. In contrast, our LES based models could reach better robustness and effectiveness on most of the hard queries with much lower complexity, therefore are better choices in terms of both effectiveness and efficiency.

6.2 TREC 2013 Results

To extensively evaluate the performance of LES, we also conduct experiments on TREC 2013 Web track data [14], which contains 50 new queries. Different from previous years' experiment setup, ClueWeb12, a newer successor of ClueWeb09, is used as the dataset for TREC 2013 Web track. Similarly, FACC1 annotation is provided as well[10]. We follow the similar experimental setup in Section 5 to process the data. In particular, we employ Waterloo Spam Rankings [16] to filter out spam documents (percentile-score threshold is set to 70 based on recommendation), apply Porter stemmer and remove stop words for both entity profile estimation and document retrieval. Since no existing entity annotation for queries in 2013 data, we manually perform entity annotation over the title fields of all 50 queries. Besides DIR, we choose LCE and KC as baselines, as they represent the best query expansion method and concept based modeling approach respectively based on the results in Section 5.2. The results of LES based models as well as baselines under five-fold cross-validation are reported in Table 8. Similarly, results of LES-COL and LES-FB are based on the re-ranking of top ranked documents of DIR. All the result evaluations are reported in nDCG@20 and ERR@20, as well as nDCG@10 and ERR@10, similar as in Section 5.

Clearly, our LES based models also demonstrate superior performance over all the baselines, and LES-FB could outperform all the baselines significantly. The better performance of LES-FB than LES-COL consistent with results in Section 6.1, as the manual entity annotations are of better quality than the

[10] http://lemurproject.org/clueweb12/FACC1/

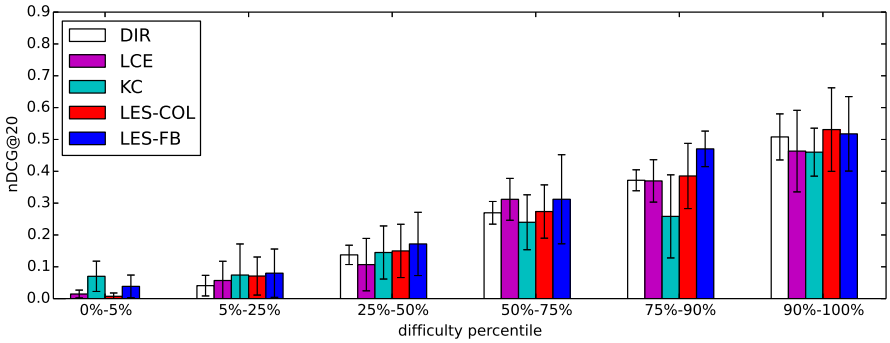


Fig. 9: Mean performance (nDCG@20) of different query difficulties on ClueWeb12 Category A data (TREC 2013). Queries are grouped based on the percentile of DIR. Error bars represent standard deviation.

automatic annotation used in Section 5.2. Similarly, we group all the queries into 6 sets based on the percentile of DIR baseline, and plot the average performance (nDCG@20) in each set in Figure 9. We could observe that both LES-COL and LES-FB demonstrate strong effectiveness across most query difficult levels. In summary, it further confirms the effectiveness and robustness of LES.

6.3 TREC 2014 Results

We participated the TREC 2014 Web track and submitted one run to the ad hoc task based on LES based on axiomatic approach with semantic term expansion [53], a very strong baseline which ranked at top 2 in TREC 2013 Web track [14]. The results [37] are very promising, as our LES based run could further improve the performance over the strong baseline significantly, and ranks at top 1 among all the 30 submitted runs [15]. Specifically, our best run is 13.2% higher than the second run and 15.6% higher than the third run in terms of ERR-IA@20 (Intent-Aware Expected Reciprocal Rank), the main evaluation measure.

We also participated the risk sensitive task. The goal is to maximize the retrieval gain of relevant documents while minimizing retrieval losses with respect to a baseline line. A system which performs better on one query will receive gain (the absolute difference of evaluation measure) as reward, and it would receive harsh penalty (α times the absolute difference in evaluation measure). In the problem setup of TREC 2014, α is set to 5. Therefore, a robust system should be able to improve more while hurt *much less* to reach good performance. We applied LES in the risk-sensitive task and ranked at top 1 among all the 12 submitted runs in terms of both ERR@20 and nDCG@20. It further demonstrates that LES is a very effective and promising approach

to improve the Web retrieval performance in terms of both effectiveness and robustness.

7 Discussions and Implications

7.1 Limitations of LES

Although experimental evaluations demonstrate that LES works empirically well on entity-bearing queries, it does not necessarily imply LES could be directly applied to replace existing ad hoc retrieval models. It is therefore worthwhile to discuss the limitations of LES and possible approaches to tackle them before we discuss the possible applications of LES.

7.1.1 Run-Time Efficiency

Efficiency would be a top concern when applying IR models to serve online retrieval requests, particularly on a large data collection. Compared to the traditional LM approach, the online computational overhead of LES consists of three parts: (1) entity annotation on query; (2) selection of top-k entities; (3) re-ranking of top-n documents. Existing off-the-shelf entity recognition toolkits could already handle the entity annotation on queries well with high accuracy and low latency. By building an inverted index of entity profiles, the overhead of entity selection process could be reduced to constant scale with common optimization techniques applied in ad hoc document retrieval. The overhead of top-n documents re-ranking could be optimized by modern distributed computing architecture as the relevance score for each document can be estimated independently. With proper handling, the computational overhead can be reduced to constant scale for each query and the system could be easily extended to large scale data collection.

7.1.2 Sensitivity on Entity Extraction Quality

Unlike traditional LM approaches which only require term features to estimate relevance score by leveraging the bag-of-words representation for queries and documents, LES requires entity annotation on queries and documents, similar to other entity-based models like EQFE model by Dalton et al. [20]. There is no doubt that the quality of entity annotation in queries is important to the performance of LES. However, LES is relatively robust against the entity annotation quality in document than other models like EQFE model, as LES does not directly rely on the entity annotation in relevant documents. Instead, LES leverages the whole collection to estimate the entity model. Given a large document collection with moderate entity annotation, the quality of entity model would be fair enough for retrieval. Besides, LES-FB does not rely on the entity annotations in documents to estimate entity model, as it directly uses entity profile from knowledge base to estimate the entity model. It means

that with the help of knowledge base, LES-FB would work well even without the entity annotation in documents, as knowledge bases like Freebase and Wikipedia could provide high quality data of entities with extensive coverage.

7.2 Possible Future Directions

As we discussed in Section 4.5, due to the fact that the characteristics of different queries vary, LES may perform well on some queries and fail on others, and we propose to predict the optimal interpolation parameter λ to balance the impacts of LES. Although empirical evaluations in Section 5.4.1 demonstrate that our proposed learning approach could effectiveness predict λ to reach robust performance, the prediction is still far from optimal and there is still a lot of room for improvements. As the optimal λ value is correlated with query difficulty, existing query performance prediction methods [32,56] could be leveraged to better predict λ based on performance prediction results and further improve the performance. A large training data set is also desirable to improve the prediction accuracy.

One of the dominant applications of LES is Web search, as a large portion of queries submitted to Web search engine bear entities [35], and they would benefit from the effectiveness and robustness of LES. One important feature for modern Web search engines is personalized search, as search engine vendors could collect the user preference from query log, click through data to provide personalized search results tailored specifically to fit an individual user's interests and preferences. LES could also be easily be adopted to personalized search, as the top-k entities served as dimensions in the latent space could be selected based on user's interests and preferences.

Another promising application of LES is question answering, which requires query parsing, relevant document retrieval and answer extraction. The relevant document retrieval is a critical step as relevant documents are the prerequisite of high-quality answer extraction. As the queries for question answering are mostly in natural language and they usually carry entities, LES could help retrieve relevant documents which are more likely to contain the answer.

8 Conclusions and Future Work

In this paper, we proposed Latent Entity Space (LES), a novel retrieval approach, which estimates document relevance in a high-dimensional entity space. Experimental results over TREC collections showed that LES can bring significant improvements over several state-of-the-art retrieval models for entity-bearing queries, demonstrating that LES is capable of capturing additional semantic content missed by existing models including Relevance Model and Latent Concept Expansion.

To our best knowledge, this is the first investigation on leveraging an entity space to model the relevance between queries and documents. It complements

the term-based retrieval models and opens up many research directions on entity-centric information retrieval. First, it would be interesting to explore more methods to estimate the query and document projection in LES. Second, it is beneficial to figure out what kind of query features would help find better related entities to further improve the performance. Finally, predicting what type of queries can benefit from the proposed LES approach would be a promising direction to pursue.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number IIS-1423002. We thank the anonymous reviewers for their useful comments.

References

1. K. Balog, L. Azzopardi, and M. De Rijke. Formal Models for Expert Finding in Enterprise Corpora. In *SIGIR*, pages 43–50, 2006.
2. K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 Entity Track. In *Proceedings of TREC*, 2010.
3. K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2010 Entity Track. In *Proceedings of TREC*, 2011.
4. M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web. In *IJCAI*, volume 7, pages 2670–2676, 2007.
5. M. Bendersky and W. B. Croft. Discovering Key Concepts in Verbose Queries. In *SIGIR*, pages 491–498, 2008.
6. B. Billerbeck and J. Zobel. Questioning Query Expansion: An Examination of Behaviour and Parameters. In *Proceedings of the 15th Australasian database conference-Volume 27*, pages 69–76. Australian Computer Society, Inc., 2004.
7. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
8. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD*, pages 1247–1250, 2008.
9. M. J. Cafarella, J. Madhavan, and A. Halevy. Web-Scale Extraction of Structured Data. *ACM SIGMOD Record*, 37(4):55–61, 2009.
10. C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 Web Track. In *TREC*, 2009.
11. C. L. A. Clarke, N. Craswell, I. Soboroff, and G. Cormack. Overview of the TREC 2010 Web Track. In *TREC*, 2010.
12. C. L. A. Clarke, N. Craswell, I. Soboroff, and E. Voorhees. Overview of the TREC 2011 Web Track. In *TREC*, 2011.
13. C. L. A. Clarke, N. Craswell, and E. Voorhees. Overview of the TREC 2012 Web Track. In *TREC*, 2012.
14. K. Collins-Thompson, P. Bennett, F. Diaz, C. L. A. Clarke, and E. M. Voorhees. TREC 2013 Web Track Overview. In *TREC*, 2013.
15. K. Collins-Thompson, C. Macdonald, P. Bennett, F. Diaz, and E. M. Voorhees. TREC 2014 Web Track Overview. In *TREC*, 2014.
16. G. Cormack, M. Smucker, and C. Clarke. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. *Information Retrieval*, 14(5):441–465, 2011.
17. C. Cortes and V. Vapnik. Support-Vector Networks. *Machine learning*, 20(3):273–297, 1995.

18. N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC 2005 Enterprise Track. In *Proceedings of TREC*, 2005.
19. S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.
20. J. Dalton, L. Dietz, and J. Allan. Entity Query Feature Expansion using Knowledge Base Links. In *SIGIR*, pages 365–374, 2014.
21. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by Latent Semantic Analysis. *JASIS*, 41(6):391–407, 1990.
22. G. Demartini. *From People to Entities: Typed Search in the Enterprise and the Web*. PhD thesis, Leibniz University of Hannover, Germany, 2011.
23. G. Demartini, A. de Vries, T. Iofciu, and J. Zhu. Overview of the INEX 2008 Entity Ranking Track. In *Focused Retrieval and Evaluation*, pages 243–252, 2009.
24. G. Demartini, J. Gaugaz, and W. Nejdl. A Vector Space Model for Ranking Entities and Its Application to Expert Search. In *ECIR*, pages 189–201, 2009.
25. 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.
26. J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and Feedback Models for Blog Feed Search. In *SIGIR*, pages 347–354, 2008.
27. H. Fang and C. Zhai. Probabilistic Models for Expert Finding. In *ECIR*, pages 418–430, 2007.
28. J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff. Building an Entity-Centric Stream Filtering Test Collection for TREC 2012. In *Proceedings of TREC*, 2012.
29. E. Gabrilovich and S. Markovitch. Wikipedia-based Semantic Interpretation for Natural Language Processing. *Journal of Artificial Intelligence Research*, 34(2):443, 2009.
30. E. Gabrilovich, M. Ringgaard, and A. Subramanya. FACC1: Freebase annotation of ClueWeb corpora, Version 1 (Release date 2013-06-26, Format version 1, Correction level 0). http://lemurproject.org/clueweb09/FACC1/, June 2013.
31. F. A. Grootjen and T. P. Van Der Weide. Conceptual Query Expansion. *Data & Knowledge Engineering*, 56(2):174–193, 2006.
32. B. He and I. Ounis. Query performance prediction. *Information Systems*, 31(7):585–594, 2006.
33. J. Lafferty and C. Zhai. Probabilistic Relevance Models Based on Document and Query Generation. *Language Modeling and Information Retrieval, Kluwer International Series on Information Retrieval*, 2003.
34. V. Lavrenko and W. B. Croft. Relevance-Based Language Models. In *SIGIR*, pages 120–127, 2001.
35. T. Lin, P. Pantel, M. Gamon, A. Kannan, and A. Fuxman. Active Objects: Actions for Entity-Centric Search. In *WWW*, pages 589–598, 2012.
36. X. Liu, F. Chen, H. Fang, and M. Wang. Exploiting Entity Relationship for Query Expansion in Enterprise Search. *Information Retrieval*, 17(3):265–294, 2014.
37. X. Liu, P. Yang, and H. Fang. Entity Came to Rescue - Leveraging Entities to Minimize Risks in Web Search. In *TREC*, 2014.
38. C. Macdonald and I. Ounis. Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. In *CIKM*, pages 387–396, 2006.
39. D. Metzler and W. B. Croft. A Markov Random Field Model for Term Dependencies,. In *SIGIR*, pages 472–479, 2005.
40. D. Metzler and W. B. Croft. Latent Concept Expansion Using Markov Random Fields. In *SIGIR*, pages 311–318, 2007.
41. D. N. Milne, I. H. Witten, and D. M. Nichols. A Knowledge-Based Search Engine Powered by Wikipedia. In *CIKM*, pages 445–454, 2007.
42. D. Petkova and W. B. Croft. Proximity-based Document Representation for Named Entity Retrieval. In *CIKM*, pages 731–740, 2007.
43. J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR*, pages 275–281, 1998.
44. J. Pound, P. Mika, and H. Zaragoza. Ad-hoc Object Retrieval in the Web of Data. In *WWW*, pages 771–780, 2010.

45. S. E. Robertson and S. Walker. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *SIGIR*, pages 232–241, 1994.
46. G. Salton, A. Wong, and C.-S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.
47. I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the TREC 2006 Enterprise Track. In *Proceedings of TREC*, 2006.
48. H. B. Styltsvig. *Ontology-based information retrieval*. PhD thesis, Roskilde University, Denmark, 2006.
49. D. Vallet, M. Fernández, and P. Castells. An Ontology-based Information Retrieval Model. In *The Semantic Web: Research and Applications*, pages 455–470. Springer, 2005.
50. L. Wang, P. N. Bennett, and K. Collins-Thompson. Robust Ranking Models via Risk-Sensitive Optimization. In *SIGIR*, pages 761–770, 2012.
51. X. Wei and W. B. Croft. LDA-Based Document Models for Ad-hoc Retrieval. In *SIGIR*, pages 178–185, 2006.
52. Y. Xu, G. J. Jones, and B. Wang. Query Dependent Pseudo-Relevance Feedback based on Wikipedia. In *SIGIR*, pages 59–66, 2009.
53. P. Yang and H. Fang. Evaluating the Effectiveness of Axiomatic Approaches in Web Track. In *TREC*, 2013.
54. C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *SIGIR*, pages 334–342, 2001.
55. C. Zhai and J. Lafferty. Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In *CIKM*, pages 403–410, 2001.
56. Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 543–550. ACM, 2007.