Exploring Summary-Expanded Entity Embeddings for Entity Retrieval

Shahrzad Naseri[1] John Foley[2] James Allan[1] Brendan T. O'Connor[1]

[1]College of Information and Computer Sciences
University of Massachusetts Amherst
{shnaseri,allan,brenocon}@cs.umass.edu

[2]Department of Computer Science
Smith College
jjfoley@smith.edu

Abstract

Entity retrieval is an important part of any modern retrieval system and often satisfies user information needs directly. Word and entity embeddings are a promising opportunity for new improvements in retrieval, especially in the presence of vocabulary mismatch problems.

We present an approach to entity embedding that leverages the summary of entity articles from Wikipedia in order to form a richer representation of entities. We present a brief evaluation using the DBPedia-Entity-v2 dataset. Our evaluation shows that our new, summary-inspired representation provides improvements over both standard retrieval and pseudo-relevance feedback baselines as well as over a straightforward word-embedding model. We observe that this representation is particularly helpful for the verbose queries in the INEX-LD and QALD-2 subsets of our test collection.

1 Introduction

Recently, knowledge cards, conversational answers, and other focused responses to user queries have become possible for most search engines. Underlying most of these answers in search engine response pages is search based on knowledge graphs and the availability of rich information for named entities. In particular, named entities such as people, organizations, or concepts are often provided as the focused response to user queries. In a study of the Yahoo web search query logs, Pound et al. [35] showed that more than 50% of the queries target specific entities or lists of entities. Since their study, more entity-focused responses have appeared in major web search engines.

Of course, rich knowledge bases play a key role in the use of entities in a search. Structured data published in knowledge bases such as DBpedia[1], Freebase[2], and YAGO[3] continue to grow in a variety of languages. In order to answer the queries directly from such knowledge bases, the *entity retrieval* task has been defined: return a ranked list of entities relevant to the user's query. This task is typically approached by finding entities with a "meaning" that is similar to the query.

Capturing that semantic ("meaning") similarity between vocabulary terms, pieces of text, and sentences has been a substantial problem in information retrieval and natural language processing (NLP), for which a wide variety of approaches have been introduced [10, 37]. The word embeddings method assigns terms a low-dimensional (compared to the vocabulary size) vector and represents vocabulary terms by capturing co-occurrence information between the terms, using a likelihood approximation of the terms' appearance within a window context. Word2vec [28] and GloVe [31] are examples of widely used word embeddings that are obtained based on a neural network-based language model and matrix factorization technique, respectively.

There has been substantial work on defining embeddings for not just single words but for entities [45, 49, 8, 46, 24], but there is no clear baseline for ranking entities with such compressed semantic representations. In fact, when trying to re-use task-specific entity embeddings for retrieval tasks, results can be less than impressive: e.g., RDF2Vec [38] was designed for data mining and has been shown to under-perform simple retrieval baselines like BM25 on more specific tasks [29]. Although fully-deep models that leverage entities exist [44], often we do not have enough data

[1] `http://dbpedia.org`

[2] `http://freebase.org`

[3] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

to train supervised embeddings.

We propose a simple entity embedding model that focuses on representing an entity based on other entities crucial to its summary. Here, we use the entities that appear inside a DBPedia abstract. Since we use links present in the abstract, these entity mentions were effectively annotated by the human authors of those articles.

In summary, we investigate the problem of entity retrieval for improving retrieval results using word and entity embeddings. We use the queries of DBpedia-Entity (v2) dataset introduced by Hasibi et al. [18] in order to evaluate our EntityVec representation on its ability to directly rank entities. We demonstrate that this is an effective representation for use in entity ranking, one that provides gains beyond those provided by single-word embeddings and query expansion.

The rest of this work is organized in the following manner: We provide some background on entity retrieval in Section 2. In Section 3 we present our approach in detail. Finally, in Section 4 we empirically validate our hypotheses and discuss conclusions in Section 6.

2 Related Work

In this section, we first introduce some prior work in entity retrieval. Then we discuss the key ideas behind the word embedding techniques whose purpose is to capture the semantic similarity between vocabulary terms.

Entities are useful for a diverse set of tasks including but not limited to academic search [45], entity disambiguation [49], entity summarization [16, 15], knowledge graph completion [46, 24], etc. We will focus our discussion on entity retrieval.

2.1 Entity Retrieval

Entity ranking is a task that focuses on retrieving entities in a knowledge base and presenting them in ranked order in response to a users' information need. This task was the focus of various benchmarking campaigns including the INEX Entity Ranking track [11], the INEX Linked Data Track [42], the TREC Entity track [41, 6, 3], the Semantic Search Challenge [7, 17], and the Question Answering over Linked Data (QALD) challenge series [25]. A common goal between all of these campaigns was to address the users' need in an entity-specific way, instead of returning documents which might contain unnecessary information. However, these campaigns focused on different tasks such as list search [3, 11], related entity finding [41] and question answering [25]. All of the datasets from those campaigns were combined into the DBPedia Entity v1 [5] and v2 [18] datasets.

2.1.1 Leveraging Knowledge Bases for Entity Retrieval

Existing methods typically study the use of type information to improve entity retrieval accuracy [4, 21, 2]. Knowledge bases are typically represented as tuples of relations, often formatted in the Resource Description Framework (RDF) triple format. As a result, entities have rich fielded information and fielded retrieval methods such as BM25F [39, 32, 20] and F-SDM [48] are especially helpful. Zhiltzov et al. in particular propose the use of *name, attribute, categories, similar entities*, and *related entities* as the fields for a fielded retrieval model [48].

To take advantage of both structured and unstructured data, Schuhmacher et al. used a learning-to-rank approach which incorporates different features of both text and entities [40]. Foley et al. expand on results for their dataset by exploring minimal knowledge-base features for use in learning-to-rank [13]. Both of these studies leverage crowd-sourced judgments of entity relevance for traditional TREC ad-hoc queries.

2.1.2 Entity Retrieval without a Knowledge Base

There have also been efforts to answer entity queries that cannot be satisfied via information in the knowledge bases due to the various ways of addressing an entity in the query. In earlier work on expert finding, entities were defined by their locations in text [1, 33]. More recently, Hong et al. [19] tried to enrich their knowledge base using linked web pages and queries from a query log. In addition, Grause et al. [14] tried to present a dynamic representation for entities by collecting different representation from a variety of resources and combine them together.

In this work, we focus on entities that *can* be found in knowledge bases.

2.2 Neural and Embedding Approaches for Entity Retrieval

As our primary direction of study for this work is toward an entity representation to improve retrieval, the most relevant efforts are those that leverage word or entity embeddings in their ranking tasks.

Word embedding techniques learn a low-dimensional vector (compared to the vocabulary size) for each vocabulary term in which the similarity between the word vectors captures the semantic as well as the syntactic similarities between the corresponding words. Word embeddings are unsupervised learning methods since they only need raw textual data without any labels. There are different methods to compute the word embeddings. One of the most popular methods is using neural networks to predict words based on the context of a text. Mikolov et al. [28] introduced word2vec that learns vector representation of words via a neural network with a single layer. Word2vec is proposed in two ways, CBOW and Skip-gram. CBOW tries to predict the word based on the context, i.e., neighboring words. Skip-gram tries to predict the context. Given the word w, it tries to predict the probability of word w' being in a fixed window of word w. Another model for learning embedding vectors is based on matrix factorization, e.g., GloVe vectors [31]. Although many variants of word embeddings exist, skipgram embeddings are quite efficient and not significantly different from other variations if tuned correctly [27, 23].

Xiong et al. propose a model for ad-hoc document retrieval that represents documents in queries in both text and entity spaces, leveraging entity embeddings in their approach [44]. However, such deep models require a significant quantity of training data to learn effective models, and our approach uses far less supervision than this direction.

Entity embeddings are also used for academic search [45], for entity disambiguation [49], for question answering [8] and for knowledge graph completion [46, 24]. The benchmark paper for TREC-CAR (Complex Answer Retrieval) determined that RDF2Vec entity embeddings [38] are not as effective as BM25 for their entity-focused paragraph ranking task [29]. Our survey of related work suggests that opportunities to customize entity vectors for ranking remain relatively unexplored.

3 Embedding-Based Entity Retrieval

Vocabulary mismatch is a long-standing problem in information retrieval. Previous work [47] has proposed to incorporate word embeddings to solve this problem. In this paper, we investigate the effect of word embeddings in entity retrieval with the goal of solving vocabulary mismatches.

Moreover, since in entity retrieval we retrieve entities instead of documents, and since most of the queries are entity centric, we learn an embedding representation for entities and explore the effect of those embeddings on entity retrieval. We hypothesize that mapping the query to the entity space and comparing with the retrieved entities will improve the retrieval results. In this section, we describe our approach to validate our hypothesis that incorporating word embeddings and entity embeddings enhances entity retrieval accuracy. We also discuss query expansion [22], an approach that also attempts to address the vocabulary gap by augmenting the query with additional related words.

3.1 General Scheme of Retrieval

Given a query, q, that targets a specific entity, our task is to return a ranked list of entities likely to be relevant. In this case, each entity is represented by a short textual description. In our experiments, for example, we used the short abstract of each entity available in DBpedia. A list of candidate entities will also be retrieved using term-based retrieval models such as query likelihood model [34], efficiently creating a large pool of candidate matches.

In our model, we try to enhance the accuracy of entity retrieval by representing queries and entities by their corresponding embedding vectors. We explore two methods to represent query and entity embedding vectors, which we refer to them as WordVec and EntityVec models.

In the WordVec model each query is represented by the average of the embedding vector of the query's terms. Entities are also represented in a similar way, by averaging over the embedding vectors of the terms in the entity's abstract. The GloVe [31] pre-trained word embedding is used for the words embedding vector in the WordVec model.

In the EntityVec model, an embedding vector for entities is learned based on the Skip-gram model implemented in gensim [36]. To learn this embedding, following the approach presented in [30], we replace the Wikipedia pages' hyperlinks (links referring to other pages, i.e., entities) with a placeholder representing the entity. Consider the following excerpt, where links to other Wikipedia articles (entities) are represented by italics:

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

The excerpt will be replaced by:

> Harry Potter is a series of `Fantasy_literature` written by British author `J._K._Rowling`. The novels chronicle the life of a young `Magician_(fantasy)`, `Harry_Potter_(character)`, and his friends `Hermione_Granger` and `Ron_Weasley`, all of whom are students at `Hogwarts`

where the link is replaced by the corresponding article's title and spaces are replaced by underscores. Now each entity in the original excerpt is considered as a single "term", and an embedding is learned based on the Skip-gram model.

Table 1: Learning corpora for WordVec and EntityVec embedding vectors

Model	Learning Corpora
WordVec	Pre-trained GloVe word embedding (6B tokens of Wikipedia + Gigawords 5)
EntityVec	Full article of Wikipedia pages pre-processed according to Section 3.1

As mentioned before, entities are represented by the abstract available in DBpedia. To also consider this representation, the final embedding of a target entity is obtained by averaging over the embedding vectors of referred entities appeared in the abstract of the target entity.

In the EntityVec model, queries are represented by the average of the embedding vectors of the entities in the query. The entities in the query are annotated using TagMe [12] mention detection tool.

For both WordVec and EntityVec the similarity between query and the document is calculated by cosine similarity between their respective embedding vectors.

The final entity retrieval score is obtained by linear interpolation of the baseline, WordVec, and EntityVec models.

Table 1 reports the learning corpora for WordVec and EntityVec models. Moreover, we summarize the final embedding vector for query and entity in table 2.

3.2 Query Expansion

In an intuitive sense, query and document embedding models solve the vocabulary mismatch problem by virtue of expanding the representation. Therefore, it makes sense to compare our work to techniques in the query-expansion literature.

Lavrenko and Croft introduce relevance modeling, an approach to query expansion that derives a probabilistic model of term importance from documents that receive high scores, given the initial query [22]. They present a number of models, but the most utilized version is RM3, which is a mixture model between the top k expansion terms and the original query. Expansion terms (t) are given the following weights derived from a set of pseudo-relevant documents D_Q a query Q:

$$w(t) = \frac{1}{Z} \sum_{d \in D_Q} P(d|Q)P(t|d)$$

Terms that occur frequently $P(t|d)$ in high-scoring documents $P(d|Q)$ are given the most weight in the expansion. The Z is merely a normalizer allowing for the weights to be turned into a probability distribution over terms that occur in the pseudo-relevant document set D_Q. This baseline is often used for comparison in entity-focused retrieval literature [9, 40, 43].

4 Experimental Setup

In this section, we introduce our experimental setup, baselines, and evaluation metrics. Next, we report and discuss our result.

4.1 Data set

Our experiments are conducted on the entity search test collection DBpedia-Entity v2 [18]. This dataset originally consists of queries gathered from the seven previous competitions with relevance judgment on entities from DBpedia version 2015-10.

For word embeddings, we used the GloVe [31] pre-trained word embedding with 300 dimensions. The word embeddings were extracted from a 6 billion token collection (the Wikipedia dump 2014 plus the Gigawords 5).

To train the entity embeddings, we used the full article of Wikipedia pages obtained from the DBpedia 2016-10 dump.

4.2 Data Processing

Retrieval results were obtained using the index built from the abstract of the entities.

We used TagMe [12] as the mention detection tool for the entities in the queries. We used the Word2Vec implementation in gensim [36] for learning entities embeddings – i.e. EntityVec. As mentioned previously, to obtain EntityVec embeddings we followed the approach outlined by Ni et al. [30] and replaced the outbound hyperlinks to Wikipedia pages with a unique placeholder token. We learn embeddings of 3.0 million entities out of 4.8 million entities in Wikipedia.

4.3 Hyperparameter Settings

The μ parameter of the language modeling approach is obtained by 2-fold cross validation over the queries. The μ parameter is chosen from the set {100, 500, 1000, 1500}. To tune the RM3 hyperparameters – i.e., the original query's weight and the number of expansion terms – we use 2-fold and 5-fold cross-validation. The original weight is changed from 0.1 to 0.9 in increments of 0.1, and the number of terms is changed from 10 to 90 in increments of 20. With the tuned parameter with 2-folds and 5-folds, RM3 for short queries did not improve over the Language model approach. We note that there were another parameter settings that *did* improve RM3 over the language model but they were not discoverable in the 2-fold or 5-fold approaches. When we report RM3 results (Table 5), we report the results for 2-fold cross-validation.

The parameters for learning the EntityVec embeddings are as follows: window-size = 10, sub-sampling = 1e-3, cutoff min-count = 0. The learned embedding

Table 2: Query and retrieved entity representations for WordVec and EntityVec models.

Model	Query	Retrieved Entity
WordVec	Average of query terms' embedding vectors	Average of embedding vectors of terms in the entity's abstract
EntityVec	Average of query entities' embedding vectors	Average of embedding vectors of referred entities in the entity's abstract

dimension is equal to 200 and it is learned based on Skip-gram model.

4.4 Evaluation Metrics

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

5 Results

In this section, we explore the results of our entity representation models atop two baselines. We look at both a standard unigram approach – language modeling (LM) [34] – and an approach built on query expansion – relevance modeling (RM3).

In Table 3, we present the results of our model on top of the LM baseline for short and verbose query subsets as well as their union. We discuss the results of our models with respect to query length in Section 5.2. This Table is the appropriate table to look at overall results of our models, particularly in the "All Queries" section. Both proposed methods outperform the baseline LM model, suggesting that there is value in both our EntityVec representation and in the more traditional WordVec query expansion. Combining the two methods yields even greater accuracy across all measures.

In Table 4, we present the results of our different models atop LM using the traditional dataset subsets inside of DBPedia-Entity-v2. Since these datasets were originally constructed for different variations of the entity ranking task, we were curious if their different query types would yield different results. We discuss the results in terms of the different styles of queries in Section 5.3.

Finally, in Table 5, we examine our approaches on top of a baseline with query expansion built-in. We discuss the results of our models on this expanded baseline in Section 5.4.

5.1 Table Notation and Significance Testing

In the result tables, relative improvements over the base retrieval models – i.e. LM and RM3 – are shown as percentages to the right of the scores. Win/Tie/Loss shows the number of queries improved, unchanged, or hurt, respectively, comparing with the base retrieval models and using the MAP measure. †, ‡, and § indicate statistical significance over the (base retrieval model), (base retrieval model)+WordVec, and (base retrieval model)+EntityVec, respectively. As mentioned earlier we use two base retrieval models (LM and RM3). The best method for each metric is marked bold.

5.2 Entity Representations for Short and Verbose Queries

We found that results were quite different for verbose queries (defined as queries longer than four terms) and short queries, so our tables are broken into three sections to reflect the overall dataset and these query-length subsets.

Based on the results in Table 3 we can see that both WordVec and EntityVec improve verbose queries more than they improve short queries (particularly measured by MAP). We speculate this could be due to short queries being more prone to ambiguity, so those better query representations are built from verbose queries where the additional words provide disambiguation and thus better matching of related entities. Also for the WordVec model, it seems that the embedding of a short query does not seem to help improve matching significantly. It is also possible that some short queries are more specific so the embedding (implicitly incorporating related words) is less important. Further analysis is needed to understand this behavior fully, but we recommend that systems that use entity representations consider using query length to select an appropriate model.

If we now look at the win/tie/loss analysis for these queries at the far right of Table 3, we can see that there are many ties. This is a result of some queries lacking entities in their description. In the current version of our model, we cannot generate an entity representation if our entity linker (TagMe, in this case) does not identify any entities in queries, so each representation is identical. Even ignoring ties, we can see that there are more wins than losses so that our vector modeling approaches are helpful when entities are identified, and the magnitude of MAP improvements is higher for EntityVec than for WordVec, even though WordVec can be used for all queries and EntityVec only changes a subset.

We further note that combining WordVec and EntityVec results in additional gains, indicating that the two methods are complementary, capturing differ-

Table 3: Effect of WordVec and EntityVec models on top of LM baseline for verbose, short queries and their union. Notation explained in Section 5.1.

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

ent aspects of entities that are each useful.

5.3 Entity Representations for Different Query Sources

When we investigate the effect of our entity vector models on different types of queries, we can see some more interesting results in Table 4. Since the queries are of such diverse types, it is not surprising to observe some variation. We see that the WordVec model does not show a significant improvement in the SemSearch-ES and QALD-2 results. Since SemSearch-ES queries are mostly ambiguous keyword queries, it is possible that the WordVec representations are not specific enough to be helpful.

5.4 Entity Representations and Query Expansion

Finally, we evaluate the proposed methods in the pseudo-relevance feedback scenario. We choose RM3 which is a state-of-the-art PRF method that has been shown to perform well in various collections [26]. Table 5 shows the results for the proposed methods and the RM3 baseline.

We observe the same kind of improvements over the RM3 baseline with our WordVec and EntityVec models that we saw on top of our keyword-query baseline. This is a really interesting observation because it shows that our embedding models are somehow orthogonal to a state-of-the-art query expansion model, which is often pointed to as the source of improvement for embedding approaches.

We note that in this dataset, the RM3 methods actually lowers the effectiveness for short queries compared to using LM alone. The WordVec and EntityVec models compensate somewhere for that reduction, but are not sufficient to recover all of the loss.

In future work, we hope to analyze the relevant entities discovered by our embedding approaches that are not present in the RM3 baselines in order to better understand where our improvements are coming from. For the EntityVec gains, we hypothesize that we have been able to encode critical information about the entity graph by modifying entity vectors to include their most important neighbors.

6 Conclusion And Future Work

In this study, we expanded on traditional entity embeddings by incorporating information from related entities that are mentioned in their summary. We demonstrated the efficacy of this model on a popular entity ranking collection in comparison to simpler word2vec style models and traditional retrieval models. In our comparison to RM3, a pseudo-relevance feedback query-expansion approach, we demonstrate that the utility of our entity modeling is not limited to query expansion – or at least, it provides a useful and novel method of query expansion in comparison to this popular approach.

In order to fully validate our model, we intend to compare it to other unsupervised and semi-supervised entity embedding representations. We hope to explore more comparisons in future work, as well as more variations of our entity embedding model.

Acknowledgement

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-1617408. Any opinions, findings and conclusions or recommendations expressed in this material

Table 4: Effect of WordVec and EntityVec models on top of LM baseline for different query types. Notation explained in Section 5.1.

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

Table 5: Effect of WordVec and EntityVec models on top of RM3 baseline for verbose, short queries and their union. Notation explained in Section 5.1.

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

are those of the authors and do not necessarily reflect those of the sponsors.

References

[1] K. Balog, L. Azzopardi, and M. De Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2006.

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

[3] K. Balog, D. Carmel, and P. Arjen. de vries, daniel m. herzig, peter mika, haggai roitman, ralf schenkel, pavel serdyukov, thanh tran duc. In *The first joint international workshop on entity-oriented and semantic search (JIWES), ACM SIGIR Forum*, volume 46, 2012.

[4] K. Balog and R. Neumayer. Hierarchical target type identification for entity-oriented queries. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2391–2394. ACM, 2012.

[5] K. Balog and R. Neumayer. A test collection for entity search in dbpedia. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 737–740. ACM, 2013.

[6] K. Balog, P. Serdyukov, and A. P. d. Vries. Overview of the trec 2010 entity track. Technical report, NORWEGIAN UNIV OF SCIENCE AND TECHNOLOGY TRONDHEIM, 2010.

[7] 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 *Proceedings of the 1st international workshop on entity-oriented search workshop (SIGIR 2011), ACM, New York*, 2011.

[8] A. Bordes, J. Weston, and N. Usunier. Open question answering with weakly supervised embedding models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer, 2014.

[9] J. Dalton, L. Dietz, and J. Allan. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM, 2014.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.

[11] G. Demartini, T. Iofciu, and A. P. De Vries. Overview of the inex 2009 entity ranking track. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 254–264. Springer, 2009.

[12] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE software*, 29(1):70–75, 2012.

[13] J. Foley, B. O'Connor, and J. Allan. Improving entity ranking for keyword queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2061–2064. ACM, 2016.

[14] D. Graus, M. Tsagkias, W. Weerkamp, E. Meij, and M. de Rijke. Dynamic collective entity representations for entity ranking. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 595–604. ACM, 2016.

[15] K. Gunaratna, K. Thirunarayan, A. Sheth, and G. Cheng. Gleaning types for literals in rdf triples with application to entity summarization. In *International Semantic Web Conference*, pages 85–100. Springer, 2016.

[16] K. Gunaratna, K. Thirunarayan, and A. P. Sheth. Faces: Diversity-aware entity summarization using incremental hierarchical conceptual clustering. In *AAAI*, pages 116–122, 2015.

[17] H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. Thompon, and D. T. Tran. Evaluating ad-hoc object retrieval. In *IWEST@ ISWC*, 2010.

[18] F. Hasibi, F. Nikolaev, C. Xiong, K. Balog, S. E. Bratsberg, A. Kotov, and J. Callan. Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1265–1268. ACM, 2017.

[19] K. Hong, P. Pei, Y.-Y. Wang, and D. Hakkani-Tür. Entity ranking for descriptive queries. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 200–205. IEEE, 2014.

[20] K. Y. Itakura and C. L. Clarke. A framework for bm25f-based xml retrieval. In *Proceedings of*

the 33rd international ACM SIGIR conference on Research and development in information retrieval, pages 843–844. ACM, 2010.

[21] R. Kaptein and J. Kamps. Exploiting the category structure of wikipedia for entity ranking. *Artificial Intelligence*, 194:111–129, 2013.

[22] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[23] O. Levy, Y. Goldberg, and I. Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[24] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187, 2015.

[25] V. Lopez, C. Unger, P. Cimiano, and E. Motta. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 21:3–13, 2013.

[26] Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1895–1898. ACM, 2009.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[29] F. Nanni, B. Mitra, M. Magnusson, and L. Dietz. Benchmark for complex answer retrieval. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, pages 293–296. ACM, 2017.

[30] Y. Ni, Q. K. Xu, F. Cao, Y. Mass, D. Sheinwald, H. J. Zhu, and S. S. Cao. Semantic documents relatedness using concept graph representation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 635–644. ACM, 2016.

[31] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[32] J. R. Pérez-Agüera, J. Arroyo, J. Greenberg, J. P. Iglesias, and V. Fresno. Using bm25f for semantic search. In *Proceedings of the 3rd international semantic search workshop*, page 2. ACM, 2010.

[33] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. *International Journal on Artificial Intelligence Tools*, 17(01):5–18, 2008.

[34] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998.

[35] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World wide web*, pages 771–780. ACM, 2010.

[36] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[37] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[38] P. Ristoski and H. Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer, 2016.

[39] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49. ACM, 2004.

[40] M. Schuhmacher, L. Dietz, and S. Paolo Ponzetto. Ranking entities for web queries through text and knowledge. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1461–1470. ACM, 2015.

[41] P. Serdyukov and A. De Vries. Delft university at the trec 2009 entity track: Ranking wikipedia entities. Technical report, DELFT UNIV OF TECHNOLOGY (NETHERLANDS), 2009.

[42] Q. Wang, J. Kamps, G. R. Camps, M. Marx, A. Schuth, M. Theobald, S. Gurajada, and A. Mishra. Overview of the inex 2012 linked data track. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[43] C. Xiong and J. Callan. Esdrank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 951–960. ACM, 2015.

[44] C. Xiong, J. Callan, and T.-Y. Liu. Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 763–772. ACM, 2017.

[45] C. Xiong, R. Power, and J. Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*, pages 1271–1279. International World Wide Web Conferences Steering Committee, 2017.

[46] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[47] H. Zamani and W. B. Croft. Embedding-based query language models. In *Proceedings of the 2016 ACM international conference on the theory of information retrieval*, pages 147–156. ACM, 2016.

[48] N. Zhiltsov, A. Kotov, and F. Nikolaev. Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–262. ACM, 2015.

[49] S. Zwicklbauer, C. Seifert, and M. Granitzer. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 425–434. ACM, 2016.