Joint Word and Entity Embeddings for Entity Retrieval from a Knowledge Graph

Fedor Nikolaev[1,2] and Alexander Kotov[1(✉)]

[1] Wayne State University, Detroit, MI 48202, USA
{fedor,kotov}@wayne.edu
[2] Kazan Federal University, Kazan, Russia

Abstract. Recent years have witnessed the emergence of novel models for ad-hoc entity search in knowledge graphs of varying complexity. Since these models are based on direct term matching, their accuracy can suffer from a mismatch between vocabularies used in queries and entity descriptions. Although successful applications of word embeddings and knowledge graph entity embeddings to address the issues of vocabulary mismatch in ad-hoc document retrieval and knowledge graph noisiness and incompleteness, respectively, have been reported in recent literature, the utility of joint word and entity embeddings for entity search in knowledge graphs has been relatively unexplored. In this paper, we propose Knowledge graph Entity and Word Embedding for Retrieval (KEWER), a novel method to embed entities and words into the same low-dimensional vector space, which takes into account a knowledge graph's local structure and structural components, such as entities, attributes, and categories, and is designed specifically for entity search. KEWER is based on random walks over the knowledge graph and can be considered as a hybrid of word and network embedding methods. Similar to word embedding methods, KEWER utilizes contextual co-occurrences as training data, however, it treats words and entities as different objects. Similar to network embedding methods, KEWER takes into account knowledge graph's local structure, however, it also differentiates between structural components. Experiments on publicly available entity search benchmarks and state-of-the-art word and joint word and entity embedding methods indicate that a combination of KEWER and BM25F results in a consistent improvement in retrieval accuracy over BM25F alone.

1 Introduction

Entity search is an information retrieval (IR) task aimed at addressing information needs focused on abstract or material objects, such as people, organizations, products and book characters. Such information needs include finding a particular entity (e.g. *"Einstein relativity theory"*), an attribute or a property of an entity (e.g. *"Who founded Intel?"*), an entity by its property (e.g. *"England football player highest paid"*) or a list of entities matching a description (e.g.

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12035, pp. 141–155, 2020.
https://doi.org/10.1007/978-3-030-45439-5_10

"Formula 1 drivers that won the Monaco Grand Prix") and can be formulated as short or "telegraphic" keyword queries or natural language questions [2,18]. Target entity or a list of entities for these information needs can be retrieved from a knowledge graph, such as Wikipedia, DBpedia, Freebase or Wikidata.

In prior research, the problem of entity retrieval from a knowledge graph was cast into a special case of structured document retrieval [21,22,25,45], database search [18,32] or a combination of the two [5,35]. Such methods take into account only the directly adjacent structural components of a knowledge graph (entities, predicates, categories, and literals) when constructing a document for an entity from a knowledge graph. As a result, a significant amount of potentially useful information can be lost by not taking into account the local structure of knowledge graphs or structural components that are separated by more than one edge in a knowledge graph. For example, the entities *Michelangelo* and *Sistine Chapel* are separated by 3 edges in DBpedia. Another drawback of existing methods is that they only consider entities as points (i.e. entity documents) in a high-dimensional space, with the number of dimensions equal to vocabulary size. This can lead to a well-known problem of vocabulary "gap" between queries and documents for relevant entities. For example, searchers looking for the entities that are related to the Musée National d'Art Moderne can pose the queries containing the terms "Beaubourg" or "MNAM", some or all of which may not be in the documents corresponding to the relevant entities. In ad-hoc document retrieval, vocabulary mismatch has been successfully addressed by employing the methods that create a low-dimensional representation of words and documents, such as Latent Semantic Indexing [8], Latent Dirichlet Allocation [38], word2vec [19], and GloVe [26]. Many recently proposed approaches [10,15,36,40,41] have successfully utilized word embeddings to address vocabulary mismatch in ad-hoc document IR.

At the same time, to address the issues of graph incompleteness and noisiness, a number of methods have been proposed for knowledge graph embedding, such as RESCAL [24], TransE [3], and NTN [33]. These methods represent knowledge graph entities and relations as vectors in the same embedding space with geometrical constraints that encode the local structure of knowledge graphs. Although these methods have been shown to be effective for the task of knowledge graph link prediction, the embedding spaces constructed by these methods as well as network embedding methods, such as DeepWalk [28], LINE [34], and node2vec [11] do not consider words and, therefore, *cannot be utilized in the tasks that involve both words and entities, such as entity search.* **However, the methods to construct joint embedding spaces for both words and entities that are effective for entity search in knowledge graph have not yet been explored**.

To address this issue, we propose **K**nowledge graph **E**ntity and **W**ord **E**mbeddings for **R**etrieval (KEWER), a novel method to create a low-dimensional representation of entities and words in the same embedding space that *takes into account both local structure and structural components of knowledge graphs.* KEWER samples random walks over a given knowledge graph and thus can be

considered as a hybrid between word and network embedding methods. Similar to word embedding methods, KEWER utilizes contextual co-occurrences as training data but differentiates between words and entities. Similar to network embedding methods, KEWER explicitly models the local structure of a knowledge graph, but unlike these methods, it takes into account various structural components of a knowledge graph, which allows us to jointly model both entities from the graph, such as DBpedia, and keyword queries.

We perform a series of experiments with KEWER to answer the following research questions: **RQ1:** How to learn joint word and entity embeddings that are effective for entity retrieval from a knowledge graph? **RQ2:** Which structural components of a knowledge graph are the most effective when learning joint entity and word embeddings for entity retrieval? **RQ3:** How does joint word and entity embeddings affect the retrieval accuracy of standard term matching based retrieval models for different types of entity search queries when they are utilized along with these models? **RQ4:** How does retrieval accuracy of the methods using joint word and entity embeddings compare with that of the methods using only word embeddings?

2 Related Work

Entity Search. Entity search approaches can be categorized into the ones that utilize structured information from knowledge graphs and the ones that do not. While earlier studies [5,32,35] heavily utilized knowledge graph's structure during retrieval, more recent studies [21,22,25,45] only use it to construct fielded entity representations, effectively casting entity search into an instance of structured document retrieval. Entity similarity information obtained from entity embeddings was successfully utilized for re-ranking the results of term-based retrieval models in [14,17,44] using a learning-to-rank approach. A publicly available benchmark for entity search based on DBpedia [16] and its more recent version [13], which provides graded relevance judgments obtained using crowdsourcing and subsequent conflict resolution by experts, are standard test collections for evaluating entity search methods.

Word Embeddings in IR. Significant research efforts in the IR community were devoted to assessing the utility of word embeddings for different IR tasks. While the initial and some of the recent works in this area directly utilize word embeddings obtained using the methods such as word2vec, several word embedding models specifically targeting IR [9,30,42] have been recently proposed. The Dual Embedding Space Model [20] utilizes embedding matrices, which correspond to the two layers of the CBOW or Skip-gram architectures, to re-rank retrieval results. Experiments with this model indicate that utilizing IN-OUT instead of IN-IN similarity between embeddings of a query and document words allows for better modeling of *aboutness* of a document with respect to a query.

Network Embeddings. Network embedding models aim to embed network nodes into a low-dimensional vector space. A common idea underlying these

methods is the adoption of the embedding methods from language modeling to sequences obtained using random walks on a given network. DeepWalk [28] is the first method that is based on this idea. DeepWalk trains the Skip-Gram architecture on sequences of vertices generated by random walks of specified length starting from each node in the network. The resulting embeddings can be used for various classification tasks, such as group labeling in social networks. Other notable network embedding methods are LINE [34], node2vec [11], and struc2vec [31].

Knowledge Graph Embeddings. Knowledge graph embeddings are a popular way to obtain low-dimensional dense representations for entities and predicates. A widely known TransE model [3] was proposed as a way to greatly reduce the number of parameters required to train the Structured Embeddings model [4] by using vector algebra. MEmbER [14] is an extension of GloVe [26] to learn conceptual spaces consisting of word and entity embeddings, in which the salient words in a given domain are associated with separating hyperplanes. Several studies [37,39,46] proposed a hybrid between entity and word embeddings by employing a loss function, which includes both a TransE-based component to model relations between entities and a word2vec-based component to model semantic relations between the words along with the third component, whose purpose is to align entity and word embeddings obtained by the first two components. In [23] authors take a different approach by learning word and entity embeddings without utilizing relations between entities from a knowledge graph and instead relying only on an unannotated corpus of text. None of the previously proposed approaches for learning joint word and entity embedding spaces were proposed specifically for entity search in a knowledge graph, and thus ignore important information, such as knowledge graph structural components.

3 Method

The primary goal of the proposed method is to learn joint word and entity embeddings that are effective for entity retrieval from a knowledge graph. The proposed method is based on the idea that a knowledge graph consists of key structural components. Structural components are loosely related to the fields of entity documents used extensively in knowledge graph entity search [21,27,45], but are defined in a more general way as a set of components of a knowledge graph that are directly or indirectly related to entities.

A given knowledge graph is formally defined as $G = \{E, R, A, C, S\}$, where $E = \{e_1, \ldots, e_{|E|}\}$ is a set of entities; R is a set of subject-predicate-object triples (s, p, o) where $s, o \in E$ are entities and p is a predicate; A is a set of triples (e, p, a) where $e \in E$ is an entity, p is a predicate, and a is a textual attribute that contains words $\{w_1, \ldots, w_k\}$ from vocabulary V; C is a set of entity-category pairs $(e, c), c \in K$; S is a combined set of entity-surface form (e, s), category-surface form (c, s), and predicate-surface form (p, s) pairs, where $s = \{n_1, \ldots, n_k\}$ is a set of word tokens in a surface form. The most commonly available surface form for an entity or category is its name or label (with $k \approx 3$).

Another example of a surface form for an entity is its anchor text. We do not use long surface forms, such as entity descriptions, in this study. The vocabulary of all distinct surface form tokens is denoted as N. We also define the following three **structural components** of a knowledge graph: **categories** (C), **literals** (A), and **predicates** ($P = \{p : (s, p, o) \in R \text{ or } (e, p, a) \in A\}$).

3.1 Knowledge Graph Entity and Word Embedding for Retrieval

To address **RQ1**, we propose KEWER, a method to jointly embed knowledge graph entities and words for entity search that takes into account the local structure of a knowledge graph, as well as its structural components. KEWER is based on a neural architecture that utilizes as input a set of sequences of word tokens and entity URIs produced by the following two-step procedure: (1) perform random walks over a knowledge graph to generate sequences consisting of structural components of a given knowledge graph (entities, predicates, attributes, and categories) of specified length t (2) randomly with probability r replace URIs in sequences resulting from random walks with their respective surface forms obtained from the same knowledge graph.

3.2 Proposed Method

In our approach, sequences generated from random walks can be viewed as short descriptions of entities that are accessed by them. For example, a random walk over DBpedia *Pierre_Curie* $\xrightarrow{spouse}$ *Marie_Curie* $\xrightarrow{knownFor}$ *Radioactivity* can be seen as a short description of Marie Curie, who was the wife of Pierre Curie and is known for discovering radioactivity. The objective that is used during training when given a current element from a sequence is to predict its surrounding context. In our example sequence, if *Marie_Curie* is the current element, the model will try to minimize the distance between embeddings of the context elements *Pierre_Curie*, *spouse*, *knownFor*, *Radioactivity* and an embedding of the current entity, *Marie_Curie*. The resemblance of this objective to the entity search task, when we need to predict target entity *Marie_Curie* from the user query such as "Who is known for her research on radioactivity and was the wife of Pierre Curie?" is a primary motivation for using random walks over a knowledge graph in the proposed method.

Random Walks Generation. Formally, the random walks are generated in the following way: starting from each entity e we generate γ random walks of length $\leq t$. Each random walk is independently generated by repeatedly following directed edges $(s, p, o) \in R$, such that the same node is not visited more than once during each random walk. If the **predicates** component is used, we add predicate-object pair (p, o) to the walk sequence, otherwise, we only add an object o. The walk procedure terminates when it either already contains t nodes or all the nodes adjacent to the current entity have already been visited. In the end, the randomly chosen attribute a of the last visited entity e, such that

$(e, p, a) \in A$ from the **literals** structural component is added to the sequence. If **categories** are considered, then the pairs $(e, c) \in C$ are treated as undirected edges during the walking procedure so that it does not need to be terminated when category nodes, which typically do not have outgoing edges, are reached.

Mixing with Surface Forms. To fulfill the need to work with user queries constructed in natural language in the typical ad-hoc entity retrieval scenario, the model should have the ability to properly embed the words that can be found in entity and category names. For that, after walks are generated, entity and category URIs are randomly replaced with probability r by their respective surface forms consisting of word tokens. If an entity or category has more than one surface form, the surface form for URI replacement is chosen uniformly at random from the set of available surface forms. In theory, this might create an issue with not utilizing all available surface forms for an entity, but in practice, this doesn't happen, since the number of generated random walks γ is typically much larger than the number of available surface forms for any given entity. If the **predicates** component is used, then the same procedure is also performed for the predicate URIs in the sequences.

Training Objective. Finally, to obtain the embeddings for words, entities, and, optionally, categories and predicates (if the corresponding knowledge graph structural components were used for sequence generation), the Skip-Gram-based model with Negative sampling [19] is trained on the resulting set of $|E|*\gamma$ random walks consisting of elements $\xi_1, \ldots, \xi_T$, where $\xi_{1\ldots T}$ are either URIs or words, and $T \geq t$ is the length of a random walk after replacement of the URIs with their surface forms. The model maximizes the probability of observing elements ξ_O from the context of the current element ξ_I by using the following objective:

$$\frac{1}{T}\sum_{i=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(\xi_{i+j}|\xi_i),\ \xi_{1\ldots T}\in\Xi,$$
$$\Xi = E \cup N \begin{cases} \cup\ K, \text{ if } \textbf{categories} \text{ are used} \\ \cup\ V, \text{ if } \textbf{literals} \text{ are used} \\ \cup\ P, \text{ if } \textbf{predicates} \text{ are used.} \end{cases} \tag{1}$$

where c is the size of the training context and the probability of observing context element $p(\xi_{i+j}|\xi_i)$ is defined using softmax as: $p(\xi_O|\xi_I) = \frac{\exp(\mathbf{v}'^{\top}_{\xi_O}\mathbf{v}_{\xi_I})}{\sum_{k=1}^{|\Xi|}\exp(\mathbf{v}'^{\top}_{\xi_k}\mathbf{v}_{\xi_I})}$. Note that each element ξ has two different IN and OUT [20] embeddings: $\mathbf{v}_\xi$ and $\mathbf{v}'_\xi$. In practice, calculating the softmax denominator of $p(\xi_O|\xi_I)$ is infeasible, and it is approximated using negative sampling.

The objective from Eq. (1) is maximized using stochastic gradient descent to learn IN and OUT embeddings of size d with derivatives estimated using back-propagation. To better utilize cross-dependencies between IN and OUT spaces [20], we use a concatenation of IN and OUT embeddings for words and OUT and IN embeddings for entities. Thus, our final embeddings are vectors of

size $d * 2$. Note that the proposed method can scale to large knowledge graphs, since all three steps of it are easily parallelizable.

3.3 Embedding-Based Entity Search

The obtained embeddings can be used to score entities with respect to a given user query in the following way. For a query Q consisting of the query terms $q_1, \ldots, q_k$, we compute embedding of the entire query $\mathbf{q}$ by calculating the weighted sum of embeddings of individual query words:

$$\mathbf{q} = \sum_{i=1}^{k} \frac{a}{p(q_i) + a} \mathbf{v}_{q_i}, \tag{2}$$

where $p(q_i)$ is a unigram probability of the query term q_i in the corpus of knowledge graph literals, and a is a free parameter [1]. The ranking score $KEWER(Q, e)$ of an entity e is then calculated as cosine similarity between entity embedding $\mathbf{v}_e$ and query embedding $\mathbf{q}$:

$$KEWER(Q, e) = \cos(\mathbf{q}, \mathbf{v}_e) \tag{3}$$

This score can be used directly to score all entities in a given knowledge graph, or used in a re-ranking scenario by combining it with traditional retrieval models such as BM25F-CA with the score $BM25F(Q, e)$ that uses term counts in the fields of a textual description of entity e. To parameterize the degree of influence of KEWER on the final ranking, its score can be multiplied by the importance weight β:

$$MM(Q, e) = \beta KEWER(Q, e) + (1 - \beta) BM25F(Q, e),\ 0 \leq \beta \leq 1 \tag{4}$$

Utilizing Entity Linking in Queries. Besides considering only words from queries, we can perform entity linking in queries to find the URIs of entities mentioned in them. For DBpedia, this can be done by using DBpedia Spotlight [7], SMAPH [6], or Nordlys toolkits [12]. After that, the embeddings of linked entities $e_1, \ldots, e_m$ are used in conjunction with the embeddings of query words to calculate the embedding of the entire query as follows:

$$\mathbf{q}_{el} = \sum_{i=1}^{k} \frac{a}{p(q_i) + a} \mathbf{v}_{q_i} + \sum_{i=1}^{m} s(e_i) \mathbf{v}_{e_i}, \tag{5}$$

where $s(e_i)$ is the entity linker's annotation score for the entity e_i. For linked entities' embeddings, we use a concatenation of IN and OUT embedding vectors.

We refer to the method that uses Eq. (5) to obtain query embedding as $\text{KEWER}_{el-tool}$, where *tool* is either *Sp* for Spotlight, *SM* for SMAPH, or *N* for Nordlys LTR method, depending on which toolkit was used for entity linking.

4 Experiments

We performed a series of experiments to answer the research questions stated in the introduction and to find the best configuration of our model by implementing KEWER and evaluating it on the DBpedia-Entity v2 dataset [13]. We implemented random walk generation ourselves and used gensim [29] for the Skip-Gram-based optimization step. The source code of the proposed method and all the baselines used in the experiments, detailed dataset construction instructions, as well as the resulting embeddings and runs are available at https://github.com/teanalab/kewer.

4.1 Dataset

DBpedia-Entity v2 collection [13] was used in all the experiments reported in this paper. Following the creators of that dataset, we used the English subset of DBpedia 2015-10 and only considered entities that have both *rdfs:label* and *rdfs:comment* predicates as our entity set E. Detailed statistics of this collection are provided in Table 1.

Table 1. Collection statistics.

Statistic	Value
#Entities $\|E\|$	4,612,277
#Categories $\|K\|$	981,499
#Predicates $\|P\|$	40,750
Avg. # of connected entities for entity	6.62
Avg. # of categories for entity	4.61
Avg. # of surface forms for entity	3.80
Avg. # of tokens in entity surface form	2.72
Avg. # of literals for entity	7.53
Avg. # of tokens in literal	2.45

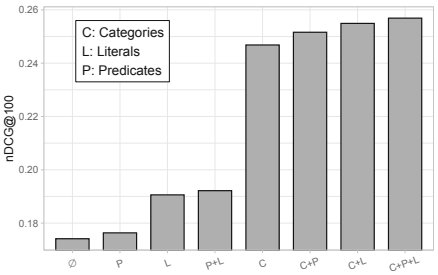

Fig. 1. nDCG_{100} for different combinations of source information used to train embeddings.

Entity search experiments were conducted using four query sets from [13]: **SemSearch ES** contains 113 named entity queries; **INEX-LD** contains 99 keyword-style IR queries; **ListSearch** contains 115 list search queries; **QALD2** contains 140 more complex question answering queries. Following DBpedia-Entity v2 creators, we mainly focus on nDCG_{100}, nDCG_{10}, and MAP evaluation metrics. The cutoff of 1000 is used for calculating MAP.

4.2 Parameter Sensitivity

To find the optimal values for the length of the random walk t and the replacement probability r, we performed a parameter sweep over the values of $t \in \{2, 3, \ldots, 10\}$ and $r \in \{0.1, 0.2, \ldots, 0.9\}$ to find out a setting that results in the highest nDCG on the query set. We found that the model always performs

better with higher values of t, and the performance saturates around $t = 10$, which we use as the parameter value in the experiments. For replacement probability, the model also performs better with higher values of r reaching top nDCG_{10} with $r = 0.9$, the value we use in the experiments. Note that we can't use $r = 1$ since there won't be any URIs for training entity embeddings ($\mathbf{v}_e$ in Eq. (3)) in this case, since they all will be replaced with surface forms. Similarly, we perform a sweep for the context size $c \in \{1, 3, \ldots, 15\}$ to find out the optimal value ($c = 5$) and term weighting parameter $a \in \{10^{-i}, 3 \times 10^{-i} : 1 \leq i \leq 5\}$ to find the optimal value ($a = 3 \times 10^{-4}$). In all subsequent experiments, we generate 100 random walks for each entity and use 5 negative samples during training.

4.3 Usefulness of Structural Components

Since it is unclear which structural components will result in embeddings that are the most useful for entity search, in the first experiment, we attempt to answer **RQ2** by trying all possible combinations of using **categories**, **literals**, and **predicates** structural components for training word and entity embeddings by KEWER. Figure 1 illustrates nDCG_{100} of KEWER averaged over all queries from four query sets when different combinations of structural components are used. In this figure, $\emptyset$ corresponds to the configuration when only R is used to generate random walks.

From Fig. 1 it can be concluded that using all three structural components is helpful for entity search, with **categories** providing the most benefit and **predicates** providing only a slight increase in retrieval accuracy. Regarding the specific query sets, we observed that using **predicates** on SemSearch ES decreased performance, which can be explained by their ineffectiveness for named entity queries that only contain entities' surface forms. On INEX-LD, we observed that not using **literals** resulted in better performance, which can be explained by the lack of attribute mentions in this query set's keyword queries. In the following experiments we use the word and entity embeddings that are trained using all three knowledge graph's structural components (**categories**, **literals** and **predicates**) and are made publicly available.

4.4 Jointly Embedding Model

As a baseline for learning embeddings, we used our implementation of the Jointly (desp) [46]. $\mathcal{L}_J$, the loss function for Jointly consists of the knowledge and the text component losses ($\mathcal{L}_K$ and $\mathcal{L}_T$, respectively) and the alignment loss $\mathcal{L}_A$:

$$\mathcal{L}_J = \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A$$

The knowledge component is formulated similar to TransE [3] with a single embedding space for entities and relations R. Both text and alignment components use textual descriptions of entities obtained from the short abstracts of entities using the *rdfs:comment* property. The text component in our implementation is formulated as a CBOW model with a single embedding space for

words. The alignment component predicts the entity embedding given the sum of embeddings of words in entity description. As an alternative to using entity descriptions, we also implemented Jointly (sf) model, where alignment and text models are trained using all available surface forms for entities from S. As in the Sect. 3.3, we define three entity linking extensions of Jointly (Jointly_{el-Sp}, Jointly_{el-SM}, and Jointly_{el-N}) using three different entity linking tools.

4.5 Entity Linking

We annotated all 467 queries using public DBpedia Spotlight API with confidence = 0.5, SMAPH, and Nordlys LTR and report the results for all entity linking models in Table 2. We don't weight linked entities by their scores in Jointly, since we have found that entity weighting is not beneficial to this model. Results indicate that using the SMAPH entity linker results in the best performance for both KEWER and Jointly. For Jointly, using entity descriptions results in better performance than using surface forms.

Table 2. Retrieval performance with entity linking. The best result is in bold.

Model	nDCG_{10}	nDCG_{100}	MAP
KEWER	0.2102	0.2569	0.1449
KEWER_{el-Sp}	0.2417	0.2803	0.1579
KEWER_{el-SM}	**0.2704**	**0.3098**	**0.1780**
KEWER_{el-N}	0.2660	0.3083	0.1775
Jointly (desp)	0.0486	0.0547	0.0211
Jointly_{el-Sp} (desp)	0.1603	0.1587	0.0838
Jointly_{el-SM} (desp)	**0.1981**	**0.1924**	**0.1014**
Jointly_{el-N} (desp)	0.1870	0.1814	0.0981
Jointly (sf)	0.0291	0.0393	0.0137
Jointly_{el-Sp} (sf)	0.1365	0.1357	0.0684
Jointly_{el-SM} (sf)	0.1685	0.1627	0.0795
Jointly_{el-N} (sf)	0.1624	0.1598	0.0836

Table 2 shows that, even without entity linking, KEWER outperforms both Jointly and Jointly with entity linking based on all metrics. A significant increase in performance of Jointly after performing entity linking suggests that word embeddings learned by Jointly are not useful for entity search, and most of its performance comes from the TransE-based component. This situation is particularly dangerous for queries that do not have entity mentions, such as "Who produced the most films?" or "What is the highest mountain?".

4.6 Mixture Model

It is clear from the above results that KEWER can be a weak ranker by itself. To achieve state-of-the-art results for ad-hoc entity search and to answer **RQ3**, KEWER can be combined with the BM25F-CA model [43], which showed good results in [13]. We implemented BM25F by indexing entities with Galago using 5 fields (*names*, *categories*, *similar entity names*, *attributes*, and *related entity names*) for entity descriptions, as was proposed in [45]. Parameters of the model were separately optimized with a coordinate ascent on each query set using nDCG_{10} as the target metric and 5 cross-validation folds from DBpedia-Entity v2. For each query, we scored the top 1000 results obtained with BM25F using $MM(Q, e)$ score from Eq. (4). The parameter β was optimized using cross-fold validation by sweeping between zero and one with 0.025 increments and

choosing the setting that results in the highest $nDCG_{100}$ on each fold's training set. BM25F results were not significantly improved by re-ranking them using Jointly, and we don't report these results. However, in our attempt to answer **RQ4**, we were able to obtain good results by applying word embeddings trained with word2vec's Skip-Gram with the hyperparameter values from Sect. 4.2, trained on the corpus of entity descriptions, where 5 aforementioned fields were combined into one textual description of an entity. The best results with word2vec for entity ranking were obtained when entity embeddings were obtained by summing up without weighting the OUT embeddings of words from their name (*rdfs:label* property), and IN embeddings were used for query terms with weighting. Results on each query set for BM25F, BM25F+word2vec, BM25F+KEWER, BM25F+KEWER$_{el-SM}$ are presented in Table 3.

Table 3. Re-ranking results per query set for KEWER with and without entity linking, and word2vec. Statistically significant improvements (determined by a randomized test with $\alpha = 0.05$) over BM25F and BM25F+word2vec are indicated by "*" and "†", respectively. The best result in each column is boldfaced.

SemSearch ES

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.6606	0.7391	0.5693
BM25F+word2vec	**0.6798***	**0.7445**	**0.5712**
BM25F+KEWER	0.6606	0.7333	0.5627
BM25F+KEWER$_{el\text{-}SM}$	0.6619	0.7409	0.5690

INEX-LD

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.4456	0.5127	0.3271
BM25F+word2vec	0.4591	0.5227	0.3406*
BM25F+KEWER	**0.4676***	**0.5298***	**0.3417***
BM25F+KEWER$_{el\text{-}SM}$	0.4577*	0.5215*	0.3363*

ListSearch

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.4287	0.4989	0.3506
BM25F+word2vec	0.4235	0.5055*	0.3551
BM25F+KEWER	0.4402†	0.5210*†	0.3752*†
BM25F+KEWER$_{el\text{-}SM}$	**0.4451***†	**0.5251***†	**0.3777***†

QALD-2

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.3442	0.4375	0.2861
BM25F+word2vec	0.3567*	0.4504*	0.2986*
BM25F+KEWER	**0.3859***†	**0.4743***†	**0.3154***†
BM25F+KEWER$_{el\text{-}SM}$	0.3800*†	0.4700*†	0.3081*†

All queries

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.4631	0.5416	0.3792
BM25F+word2vec	0.4730*	0.5504*	0.3874*
BM25F+KEWER	**0.4831***†	**0.5602***†	**0.3955***†
BM25F+KEWER$_{el\text{-}SM}$	0.4807*†	0.5601*†	0.3944*†

The results demonstrate that re-ranking by KEWER is particularly useful for complex question answering queries from QALD-2, list queries from ListSearch, and keyword queries from INEX-LD, while being less useful for simple named entity queries from SemSearch ES, where word2vec thrives. For queries from ListSearch, KEWER is particularly useful when used in combination with entity linker, while for QALD-2 and INEX-LD using entity linking provides lower performance gain. This can be explained by the lack of useful entity mentions in QALD-2 and INEX-LD queries. In QALD-2 queries, mentioned entities are

often of a different category than the entity of user's interest and have a complex relationship with it. Using the embeddings of linked entities, in this case, would skew results in the wrong direction. Instead, using plain KEWER helps to clarify the query's intent directly from its keywords.

4.7 Success/Failure Analysis

To illustrate the positive effect of using KEWER on retrieval accuracy, we analyze a sample query SemSearch_LS-50 "wonders of the ancient world" where employing KEWER embeddings resulted in a performance boost. The top results for BM25F and KEWER (without interpolation with BM25F) are presented in Table 4. From these results, it is evident that BM25F failed to capture the conceptual focus of the query by using term matching and most of its top results are only marginally relevant to the query's main focus. On the other hand, KEWER correctly identified the query's main focus on the ancient world, providing five highly relevant results in the ranking.

Table 4. Top 10 ranked entities for the query "wonders of the ancient world" for different models. Relevant results are *italicized* and highly relevant results are **boldfaced**.

BM25F	KEWER
Seven Wonders of the Ancient World	**Colossus of Rhodes**
7 Wonders of the Ancient World (video game)	**Statue of Zeus at Olympia**
Wonders of the World	**Temple of Artemis**
Seven Ancient Wonders	List of archaeoastronomical sites by country
The Seven Fabulous Wonders	**Hanging Gardens of Babylon**
The Seven Wonders of the World (album)	Antikythera mechanism
Times of India's list of seven wonders of India	Timeline of ancient history
Lighthouse of Alexandria	*Wonders of the World*
7 Wonders (board game)	*Lighthouse of Alexandria*
Colossus of Rhodes	**Great Pyramid of Giza**

An example of a query where KEWER was unable to identify query focus is "goodwill of michigan", where it returns entities that are related to Goodwill Games instead of Goodwill Industries. This is caused by the fact that there exist a lot of entities with words "Goodwill Games" in their surface forms, which makes the model believe that token "goodwill" has a strong association with games.

5 Conclusion

This paper proposed KEWER, a method to learn joint word and entity embeddings that was experimentally shown to be effective for entity search, which addresses **RQ1**.

To answer **RQ2**, we compared the effectiveness of embeddings trained on various combinations of knowledge graph structural components and found out that using a combination of **categories**, **literals**, and **predicates** results in the highest retrieval accuracy on DBpedia-Entity v2.

To answer **RQ3** and **RQ4**, we performed an evaluation of KEWER in the re-ranking scenario where it was used in combination with the BM25F retrieval model. Experimental results indicate that KEWER is particularly suitable for improving the ranking of results for complex entity search queries, such as question answering, list search, and keyword queries.

References

1. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: ICLR 2017 (2017)
2. Blanco, R., et al.: Entity search evaluation over structured web data. In: SIGIR 2011 (2011)
3. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS 2013. pp. 2787–2795 (2013)
4. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: AAAI 2011, pp. 301–306 (2011)
5. Ciglan, M., Nørvåg, K., Hluchý, L.: The semsets model for ad-hoc semantic list search. In: WWW 2012, pp. 131–140 (2012)
6. Cornolti, M., Ferragina, P., Ciaramita, M., Rüd, S., Schütze, H.: A piggyback system for joint entity mention detection and linking in web queries. In: WWW 2016, pp. 567–578 (2016). https://doi.org/10.1145/2872427.2883061
7. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Proceedings of the 9th International Conference on Semantic Systems, I-SEMANTICS 2013, pp. 121–124 (2013). https://doi.org/10.1145/2506182.2506198
8. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS **41**(6), 391–407 (1990)
9. Diaz, F., Mitra, B., Craswell, N.: Query expansion with locally-trained word embeddings. In: ACL 2016 (2016)
10. Embedding-based Query Expansion for Weighted Sequential Dependence Retrieval Model. https://doi.org/10.1145/3077136.3080764
11. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: KDD 2016, pp. 855–864 (2016). https://doi.org/10.1145/2939672.2939754
12. Hasibi, F., Balog, K., Garigliotti, D., Zhang, S.: Nordlys: a toolkit for entity-oriented and semantic search. In: SIGIR 2017, pp. 1289–1292 (2017). https://doi.org/10.1145/3077136.3084149
13. Hasibi, F., et al.: DBpedia-entity v2: a test collection for entity search. In: SIGIR 2017, pp. 1265–1268 (2017). https://doi.org/10.1145/3077136.3080751
14. Jameel, S., Bouraoui, Z., Schockaert, S.: MEmbER: max-margin based embeddings for entity retrieval. In: SIGIR 2017, pp. 783–792 (2017). https://doi.org/10.1145/3077136.3080803
15. Kuzi, S., Shtok, A., Kurland, O.: Query expansion using word embeddings. In: CIKM 2016, pp. 1929–1932 (2016). https://doi.org/10.1145/2983323.2983876

16. Lehmann, J., et al.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semant. Web **6**(2), 167–195 (2015). https://doi.org/10.3233/SW-140134
17. Liu, Z., Xiong, C., Sun, M., Liu, Z.: Explore entity embedding effectiveness in entity retrieval. In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) CCL 2019. LNCS (LNAI), vol. 11856, pp. 105–116. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_9
18. Lopez, V., Unger, C., Cimiano, P., Motta, E.: Evaluating question answering over linked data. Web Semant. **21**, 3–13 (2013). https://doi.org/10.1016/j.websem.2013.05.006
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS 2013, pp. 3111–3119 (2013)
20. Mitra, B., Nalisnick, E.T., Craswell, N., Caruana, R.: A dual embedding space model for document ranking. CoRR abs/1602.01137 (2016). http://arxiv.org/abs/1602.01137
21. Neumayer, R., Balog, K., Nørvåg, K.: On the modeling of entities for ad-hoc entity search in the web of data. In: Baeza-Yates, R., et al. (eds.) ECIR 2012. LNCS, vol. 7224, pp. 133–145. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28997-2_12
22. Neumayer, R., Balog, K., Nørvåg, K.: When Simple is (more than) good enough: effective semantic search with (almost) no semantics. In: Baeza-Yates, R., de Vries, A.P., Zaragoza, H., Cambazoglu, B.B., Murdock, V., Lempel, R., Silvestri, F. (eds.) ECIR 2012. LNCS, vol. 7224, pp. 540–543. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28997-2_59
23. Newman-Griffis, D., Lai, A.M., Fosler-Lussier, E.: Jointly embedding entities and text with distant supervision. In: Rep4NLP@ACL 2018, pp. 195–206 (2018)
24. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing YAGO: scalable machine learning for linked data. In: WWW 2012, pp. 271–280 (2012). https://doi.org/10.1145/2187836.2187874
25. Nikolaev, F., Kotov, A., Zhiltsov, N.: Parameterized fielded term dependence models for ad-hoc entity retrieval from knowledge graph. In: SIGIR 2016, pp. 435–444 (2016). https://doi.org/10.1145/2911451.2911545
26. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP 2014, pp. 1532–1543 (2014)
27. Pérez-Agüera, J.R., Arroyo, J., Greenberg, J., Iglesias, J.P., Fresno, V.: Using BM25F for semantic search. In: SEMSEARCH 2010, pp. 2:1–2:8 (2010). https://doi.org/10.1145/1863879.1863881
28. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: KDD 2014, pp. 701–710 (2014). https://doi.org/10.1145/2623330.2623732
29. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: NLP Frameworks at LREC 2010, pp. 45–50. May 2010
30. Rekabsaz, N., Mitra, B., Lupu, M., Hanbury, A.: Toward incorporation of relevant documents in word2vec. CoRR abs/1707.06598 (2017). http://arxiv.org/abs/1707.06598
31. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: Struc2vec: learning node representations from structural identity. In: KDD 2017, pp. 385–394 (2017). https://doi.org/10.1145/3097983.3098061

32. Shekarpour, S., Ngonga Ngomo, A.C., Auer, S.: Question answering on interlinked data. In: WWW 2013, pp. 1145–1156 (2013). https://doi.org/10.1145/2488388.2488488
33. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: NIPS 2013, pp. 926–934 (2013)
34. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: WWW 2015, pp. 1067–1077 (2015). https://doi.org/10.1145/2736277.2741093
35. Tonon, A., Demartini, G., Cudré-Mauroux, P.: Combining inverted indices and structured search for ad-hoc object retrieval. In: SIGIR 2012, pp. 125–134 (2012). https://doi.org/10.1145/2348283.2348304
36. Vulić, I., Moens, M.F.: Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In: SIGIR 2015, pp. 363–372 (2015)
37. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph and text jointly embedding. In: EMNLP 2014, pp. 1591–1601 (2014)
38. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: SIGIR 2006, pp. 178–185 (2006). https://doi.org/10.1145/1148170.1148204
39. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: AAAI 2016, pp. 2659–2665 (2016)
40. Zamani, H., Croft, W.B.: Embedding-based query language models. In: ICTIR 2016, pp. 147–156 (2016). https://doi.org/10.1145/2970398.2970405
41. Zamani, H., Croft, W.B.: Estimating embedding vectors for queries. In: ICTIR 2016, pp. 123–132 (2016). https://doi.org/10.1145/2970398.2970403
42. Zamani, H., Croft, W.B.: Relevance-based word embedding. In: SIGIR 2017, pp. 505–514 (2017). https://doi.org/10.1145/3077136.3080831
43. Zaragoza, H., Craswell, N., Taylor, M.J., Saria, S., Robertson, S.E.: Microsoft Cambridge at TREC 13: Web and hard tracks. In: TREC 2004 (2004)
44. Zhiltsov, N., Agichtein, E.: Improving entity search over linked data by modeling latent semantics. In: CIKM 2013, pp. 1253–1256 (2013). https://doi.org/10.1145/2505515.2507868
45. Zhiltsov, N., Kotov, A., Nikolaev, F.: Fielded sequential dependence model for ad-hoc entity retrieval in the web of data. In: SIGIR 2015, pp. 253–262 (2015). https://doi.org/10.1145/2766462.2767756
46. Zhong, H., Zhang, J., Wang, Z., Wan, H., Chen, Z.: Aligning knowledge and text embeddings by entity descriptions. In: EMNLP 2015, pp. 267–272 (2015)