Knowledge Graph Entity Representation and Retrieval

Alexander Kotov

Wayne State University, Detroit, USA
kotov@wayne.edu

Abstract. Recent studies indicate that nearly 75% of queries issued to Web search engines aim at finding information about entities, which are material objects or concepts that exist in the real world or fiction (e.g. people, organizations, products, etc.). Most common information needs underlying this type of queries include finding a certain entity (e.g. "Einstein relativity theory"), a particular attribute or property of an entity (e.g. "Who founded Intel?") or a list of entities satisfying a certain criteria (e.g. "Formula 1 drivers that won the Monaco Grand Prix"). These information needs can be efficiently addressed by presenting structured information about a target entity or a list of entities retrieved from a knowledge graph either directly as search results or in addition to the ranked list of documents. This tutorial provides a summary of the recent research in knowledge graph entity representation methods and retrieval models. The first part of this tutorial introduces state-of-the-art methods for entity representation, from multi-fielded documents with flat and hierarchical structure to latent dimensional representations based on tensor factorization, while the second part presents recent developments in entity retrieval models, including Fielded Sequential Dependence Model (FSDM) and its parametric extension (PFSDM), as well as entity set expansion and ranking methods.

1 Introduction

Search engine users often try to find concrete or abstract objects (e.g. people, organizations, scientific papers, products, etc.) rather than documents and are willing to express such information needs more elaborately than with a few keywords [13]. In particular, according to the recent studies [37, 51, 69], 3 out of every 4 queries submitted to the Web search engines either contain entity mentions or aim at finding information about entities. Most of these queries fall into the following four major categories:

1. **Entity search queries**: queries aimed at finding a specific entity either by its name (e.g. *"Ben Franklin"*, *"Einstein relativity theory"*) or description (e.g. *"England football player highest paid"*);
2. **Entity attribute queries**: queries aimed at finding an attribute or property of a given entity (e.g. *"mayor of Berlin"*);

3. **List search queries**: descriptive queries aimed at finding multiple entities (e.g. *"U.S. presidents since 1960"*, *"animals lay eggs mammals"*, *"Formula 1 drivers that won the Monaco grand prix"*);
4. **Questions**: natural language questions aimed at finding particular entities (e.g. *"for which label did Elvis record his first album?"*), entity attributes (e.g. *"what is the currency of the Czech republic"*) or relations between entities (e.g. *"which books by Kerouac were published by Viking Press?"*).

The information needs underlying such queries are much more efficiently addressed by directly presenting the target entity or a list of entities (potentially, along with an entity card containing entity image and/or short description) than a traditional ranked list of documents, which contain mentions of these entities. Implementing this functionality in search systems requires comprehensive repository of information about entities as well as the methods for retrieving and ranking entities in response to keyword queries. In this tutorial, we focus on entity information repositories in the form of knowledge graphs.

1.1 Knowledge Graphs

Recent successes in the development of Web-scale information extraction methods have resulted in the emergence of a number of large-scale entity-centric information repositories, such as DBpedia[1], Freebase[2] and YAGO[3]. These repositories adopt a simple data model based on subject-predicate-object (SPO) triples, in which a subject is always an entity, an object is either another entity, string literal or a number and a predicate designates the type of relationship between subject and object (e.g. bornIn, hasGenre, isAuthorOf, isPCmemberOf etc.). An entity is typically designated by a Uniform Resource Identifier (URI) (e.g. a URL in the case of DBpedia, which can be used to look up aggregated structured information about each entity on-line) and can be any concept that exists in the real world or fiction (e.g. person, book, color, etc.). A large number of SPO triples can be conceptualized as a directed labeled multi-graph (often referred to as a *knowledge graph*), in which the nodes correspond to entities and the edges denote typed relationships between entities.

Entities can be linked to other entities in different knowledge bases (e.g. an entity in DBpedia can be linked to an entity in Freebase). Cross-linked entities in DBpedia, Freebase and YAGO form the core of Linked Open Data (LOD) cloud[4] (also referred to as the Web of Linked Data or the Web of Data [40]), a giant distributed knowledge graph. As of 2014, the LOD cloud consisted of over 60 billion RDF triples in over 1000 interlinked knowledge graphs representing a wide range domains, from media and entertainment to e-government and science. The LOD cloud is continuing to grow as more and more Web resources are providing

[1] `http://dbpedia.org`
[2] `http://freebase.com`
[3] `http://yago-knowledge.org`
[4] `http://lod-cloud.net`

linked meta-data records in the form of RDF triples along with the traditional human-readable textual content.

1.2 Entity retrieval from knowledge graphs

The scale and diversity of knowledge stored in the Web of Data and the entity centric nature of knowledge graphs makes them perfectly suited for addressing information needs aimed at finding entities rather than documents. This scenario gives rise to *Ad-hoc Entity Retrieval from Knowledge Graphs* (ERKG). ERKG is a *unique* and *challenging* Information Retrieval (IR) problem. In particular, ERKG gives rise to two fundamental research questions:

1. **Designing entity representations that capture important aspects of semantics of both the entities and their relations to other entities:** although ERKG is similar to traditional ad-hoc document retrieval or Web search in that it assumes unstructured textual queries, a fundamental difference between these two retrieval tasks is that the units of retrieval in case of ERKG are structured objects (entities) rather than documents and the "collection" is one or several knowledge graphs. While the structure of knowledge repositories is perfect for answering structured queries based on graph patterns, it is not suitable for keyword queries. This results in additional challenges related to creating entity representations that are suitable for traditional IR models;
2. **Developing accurate and efficient retrieval models:** since ERKG involves matching unstructured queries with relevant structured objects, query understanding in this scenario involves not only accurate recognition of the key query concepts (terms and phrases) and determining their relative importance, but also matching these concepts with the correct elements of structured entity semantics of relevant entities encoded in knowledge graphs.

Next, we provide a brief overview of the recent work in entity retrieval from documents, entity retrieval using structured queries over triple stores and retrieval from graph databases, the three research directions most closely related to ERKG, as well as outline the relations of ERKG to other IR tasks.

2 History and relation to other retrieval tasks

ERKG is historically related to several other information access scenarios involving entities, graphs and knowledge graphs, such as entity retrieval, retrieval from RDF triple stores using structured queries and retrieval from graph databases.

2.1 Entity retrieval

Before the emergence and widespread popularity of knowledge graphs, several evaluation initiatives within TREC and INEX conferences introduced the problem of ad-hoc entity retrieval [9], which is focused on retrieving named entities *embedded in documents*. Entity retrieval tasks introduced by these initiatives varied from *retrieving Wikipedia articles that match a keyword query* to *retrieving named entities embedded in textual documents or Web pages*.

The Entity track at TREC 2009-2011 [10, 12] featured *related entity finding* and *entity list completion* tasks. The goal of the related entity finding task [18, 57] is to retrieve and rank related entities given a structured XML query specifying an input entity, the type of related entities and the relation between the input and related entities in the context of a given document collection. Expert finding [7] is a special case of related entity finding task in the context of enterprise search, which assumes specific types of relations (e.g. expert in), as well as specific input (e.g. area of expertise) and related entities (e.g. employee). The goal of entity list completion task is to find entities with common properties given some example entities.

The Entity Ranking track at INEX 2008-2010 (INEX-XER) [26, 28, 29] featured similar tasks, with the key difference that specific type of the target entities, rather than specific relations between the target and input entities was provided. The goal of the entity ranking task is to return a ranked list of entities, in which each entity is associated with a Wikipedia page and a set of Wikipedia categories designating the entity type, given a structured XML query that consists of the query keywords along with a set of target entity Wikipedia categories. Besides the text of Wikipedia articles, the methods proposed to address this task [8, 9, 45, 44, 27], leveraged diverse metadata provided by Wikipedia, such as categories, disambiguates and link structure.

The problem of entity retrieval has also been studied in the context of Web search. Cheng et al. [21] and Nie et al. [64] proposed language modeling-based methods to retrieve *Web objects*, which are units of information about people, products, locations and organizations extracted and aggregated from different Web sources. Guo et al. [37] proposed a probabilistic approach based on weakly supervised topic model to detect named entities in queries and identify their most likely categories (e.g. “book”, “movie”, “music”, etc.) . A method to automatically identify and display relevant actions for actionable Web search queries (e.g. show exact address and a map for a query “sea world location”) was proposed by Lin et al. [51].

2.2 Structured queries over triple stores

Information in knowledge graphs, stored in RDF triple stores, can also be accessed using structured query languages, such as SPARQL Protocol and Recursive Query Language (SPARQL). SPARQL queries consist of RDF triples with parameters and correspond to knowledge graph patterns. Since their results are typically unranked and consist of subgraphs of a knowledge graph that exactly match query patterns, SPARQL queries often fall short of satisfying the users’ information needs by returning too many or too few results. Furthermore, in order to be properly utilized, structured query languages require knowledge of the schema of a given knowledge repository and a certain level of technical skills, which many ordinary users are unlikely to possess. Several approaches to question answering over linked data translate natural language questions into SPARQL queries [77, 75, 81]. A language modeling-based method for ranking the results of structured SPARQL queries over RDF triple stores proposed by Elbassuoni et al. [31] first constructs language models (LMs) of both the query and each sub-graph in query results and then ranks the results based on the Kullback-Leibler divergence between their corresponding LMs and the query LM. Elbassuoni and Blanco [30] proposed a method for keyword search over RDF graphs, which represents RDF triples as documents and returns a ranked list of RDF subgraphs formed by joining the triples retrieved by individual query keywords.

2.3 Retrieval from graph databases

Methods for searching graph databases using structured queries [82] as well as keyword search in relational and graph databases have been extensively studied in the database community. However, these scenarios are quite different from ERKG, since keyword search over relational and graph databases returns a ranked list of non-redundant Steiner trees [2, 43, 54, 1, 41, 33] or sub-graphs [50], which contain the occurrences of query keywords. Ranking models in graph database retrieval typically leverage the graph structure by aggregating the weights of nodes and edges [1], attribute-value statistics [20] or a combination of these properties with content-based relevance measures from IR, such as TF-IDF weights [23, 21, 41, 50], probabilistic [20] or language models [64] as well as term proximity [33].

2.4 ERKG and other IR tasks

ERKG can be combined with [39] or used as an alternative to entity linking [38], which identifies the mentions of KG entities in a query, in the methods that utilize knowledge graphs to improve general purpose [25, 56, 74, 79, 80] and domain-specific [5] ad-hoc document retrieval. Term and concept graphs, such as ConceptNet [55], are special cases of knowledge graphs, in which the nodes are words or phrases and the weighted edges represent the strength of semantic relationship between them. This type of knowledge graphs was also shown to be effective at improving ad-hoc document retrieval [3, 4, 6, 47, 48].

3 Architecture of ERKG systems

Architecture of an ERKG system, an example of which is shown in Figure 1, is typically a pipeline that consists of entity retrieval, entity set expansion and entity ranking components. As can be seen in Figure 1, an ERKG system creates a structured or unstructured *textual representation* (i.e. *entity document*) for each entity in the knowledge graph (different entity representation schemes are discussed in detail in Section 4) and maintains an inverted index mapping terms to the fields of entity documents. In the first stage of the pipeline, an inverted index is used to retrieve an initial set of entities using structured document retrieval models (discussed in detail in Section 5). An initial set of entities can be expanded in the second stage of the pipeline by traversing the knowledge graph to include related entities (specific methods are discussed in Section 6). Finally, an initial set of entities along with the entities in the expanded set are ranked using learning-to-rank methods (discussed in detail in Section 7) in the last stage of the pipeline.

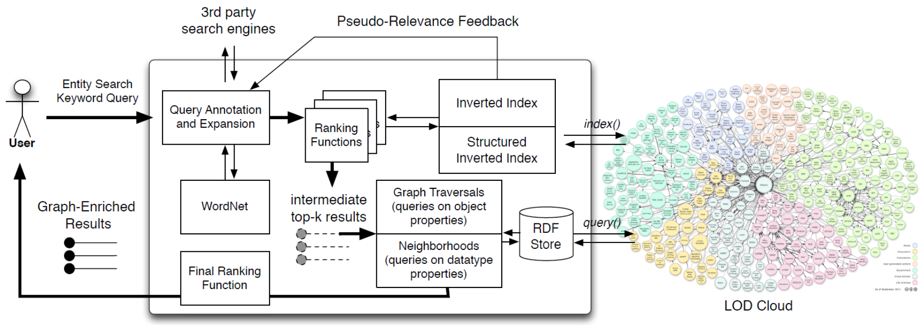


Fig. 1: **Architecture of a typical ERKG system (adopted from [76]).**

4 Entity representation

All ERKG methods working with *unstructured queries* that have been proposed to this date involve a preprocessing step, in which an entity document is built for each entity in the knowledge graph. Entity document aggregates information from all triples, in which the entity is either a subject or an object. Figure 2 illustrates this process.

Since the semantics of entities is encapsulated in the fragment of a knowledge graph around them (i.e. related entities and literals as well as predicates connecting them), it is natural to represent KG entities as structured (multi-field) documents. In the simplest entity representation method, each distinct predicate corresponds to one field of an entity document. In this case, each field of entity document consists of other entity names and literals connected to a given entity with a predicate that corresponds to this field. Since field importance weights are the key parameters of all existing models for structured document retrieval, optimization of such models for structured entity documents, which have as many fields as there are distinct predicates, would be infeasible due to prohibitively large amounts of the required training data.

To create entity documents with manageable number of fields, methods for predicate folding, or grouping predicates into a small set of predefined categories corresponding to the fields of entity representations, have been proposed. Neumayer and Balog [11, 62] proposed to represent entities as documents with two

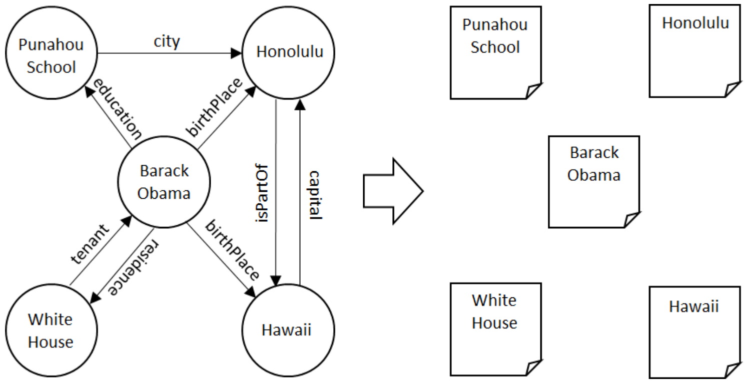


Fig. 2: **Creating documents for entities in a fragment of the knowledge graph.**

fields: title and content. The title field consists of entity names and literals that are the objects of the predicates ending with “name”, “label” or “title”, while the content field combines the objects of 1000 most frequent predicates. This simple approach combined with boosting of entities from high-quality sources, such as Wikipedia, demonstrated good results for entity search. Zhiltsov and Agichtein [83] proposed to aggregate entity names and literals in the object position in two separate fields (attributes and outgoing links). The resulting entity documents consist of 3 fields: *names* (which is similar to the *title* field in [62]), *attributes*, and *outgoing links*. This entity representation is also effective for entity search, since it allows to find entities using their attributes and relations to other entities as queries.

Structured Entity Model [61] creates entity documents with 4 fields (*name*, *attributes*, *outgoing relations* and *incoming relations*), an example of which is shown in Figure 3, while Hierarchical Entity Model [61] combines the advantages of predicate weighting and predicate folding by organizing the predicates into a two-level hierarchy of fields. The fields at the top level of the hierarchy correspond to predicate types, while the fields at the bottom level correspond to individual predicates. This scheme allows to condition the importance of a given predicate on its type and associated entity in different ways (e.g. by setting the weight of a predicate field proportional to its length or predicate popularity).

Zhiltsov et al. [84] proposed a refinement of a 3-field entity document [83] by adding the *categories* and *similar entity names* (names of entities that are subjects of `owl:sameAs` predicate with the given entity as an object) fields. The resulting entity representations with 5 fields (names, attributes, categories, similar entity names and related entity names) has been shown to be effective for entity search, list search and question answering [65, 84], since it allows to find sets of entities using one or several categories they belong to as queries, in addition to finding entities by their aliases, attributes and relations to other entities. An alternative entity document scheme with 5 fields (*text*, *title*, *object*, *inlinks*, and *type*) has been proposed by Pérez-Agüera et al. [67].

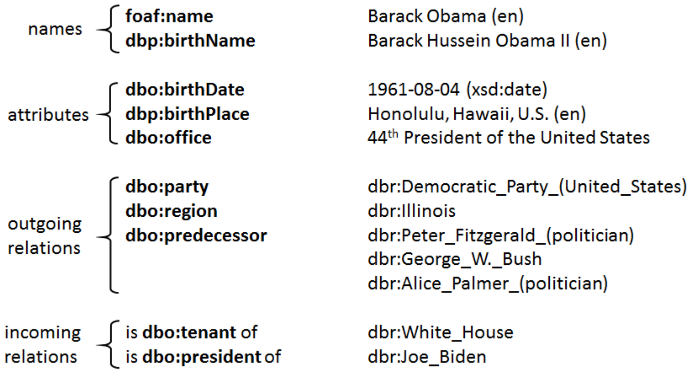


Fig. 3: **Folding predicates corresponding to entity names, attributes, outgoing and incoming links into a 4-field entity document using the approach in [61] for DBpedia entity `http://dbpedia.org/resource/Barack_Obama`.**

A major limitation of the above methods is that they create *static* entity representations, which disregard two fundamental properties of entities. The first property is that the same entity can appear in different contexts over time (e.g. entity `Germany` should be returned for queries related to World War II as well as 2014 Soccer World Cup). The second property is that entity documents change over time (e.g. entity document `Ferguson, Missouri` before and after August 2014). To take into account these two properties of entities, Graus et al. [35] proposed to leverage collective intelligence provided by different sources (e.g. tweets, social tags, query logs) to dynamically update structured entity document and tweak the weights of the fields of those documents, which correspond to different sources of entity description terms, over time. They found out that incorporating a variety of sources in creating dynamic entity descriptions allows to account for changes in entity representations over time and that social tags are the best performing single entity description source.

5 Entity retrieval

With implicit structure of keyword queries and explicit structure of entity representations, it is natural to assume that the accuracy of entity retrieval depends on the correctness of matching query concepts with different aspects of semantics of relevant entities encoded in their structure. Ambiguity of natural language can lead to many plausible interpretation of a keyword query, which combined with many possible projections of those interpretations onto structured entity representations makes ERKG a challenging IR problem.

While models for retrieving entities from knowledge graphs is the first and most important stage in the pipelines for many entity retrieval tasks, these models can also play an important role in other information seeking contexts:

1. they can be used in search systems to allow users to pose complex keyword queries in order to access and interact with structured knowledge in knowledge graphs and the Web of Data. The main advantage of keyword-based entity search systems is that they generally do not require users to master complex query languages or understand the underlying schema of a knowledge graph to be able to interact with it;
2. they can be used to retrieve more accurate and complete initial set of entities for complex and exploratory entity-centric information needs. This initial set of entities can be further expanded and/or re-ranked using task-specific approaches. Alternatively, models for ERKG can pinpoint entities of interest as the starting points for further interactive exploration of information needs and knowledge graphs [49, 60];
3. they can be used to supplement the search results obtained using document retrieval models (e.g. Web search results) with structured knowledge for the same keyword query [36, 73]. Therefore, ERKG can be considered as a separate search vertical.

Despite their potentially wide applicability, models that are designed specifically for entity retrieval from knowledge graphs have received limited attention from IR researchers. As a result, until recently, ERKG methods had to rely either on bag-of-words models [11, 61, 62, 76, 83] or on models incorporating term dependencies to retrieve structured entity documents for keyword queries.

5.1 Bag-of-words models for structured document retrieval

Mixture of Language Models (MLM) [66] and BM25F [71], the most popular bag-of-words retrieval models for structured document retrieval, are extensions of probabilistic (BM25 [70]) and language modeling-based (Query Likelihood [68]) retrieval models to structured documents, respectively. These models are based on the idea that fields in entity documents encode different aspects of relevance, but propose different formalizations of this idea. BM25F calculates the values of standard retrieval heuristics (term frequency, document length) as a linear combination of their values in different document fields and plugs these values directly into BM25 retrieval formula to obtain a retrieval score for the entire document. Robertson and Zaragoza [71] demonstrated that this strategy is superior to simple aggregation of BM25 retrieval scores for individual document fields. MLM, on the other hand, creates a language model for a structured document as a linear combination of language models for individual document fields. Probabilistic Retrieval Model for Semistructured Data (PRMS) [46] learns a simple statistical relationship between the intended mapping of query terms and their frequency in different document fields. Robust estimation of this relationship, however, requires query terms to have a non-uniform distribution across document fields and is negatively affected by sparsity when structured documents have a large number of fields. For this reason, PRMS performs relatively well on collections of documents with a small number of medium to large-size fields (e.g. movie reviews), but exhibits a dramatic decline in performance on structured documents with large number of small fields.

The key limitation of all bag-of-words retrieval models is that they do not account for the dependencies between query terms (i.e. query phrases) and are unable to differentiate the relative importance of query terms and phrases.

5.2 Retrieval models incorporating term dependencies

Markov Random Field (MRF) retrieval model [58] provided a theoretical foundation for incorporating term dependencies in the form of ordered and unordered bigrams into retrieval models. MRF considers a query as a graph of dependencies between the query terms and between the query terms and the document. MRF calculates the score of each document with respect to a query as a linear combination of potential functions, each of which is computed based on a document and a clique in the query graph. Sequential Dependence Model (SDM), the most popular variant of the Markov Random Field model (shown in Figure 4), assumes sequential dependencies between the query terms and uses three potential functions: the one that is based on unigrams and the other two that are based on bigrams, either as ordered sequences of terms or as terms co-occurring within a window of the pre-defined size. This parametrization results in the following retrieval function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} f_T(q_i, D) \; + \lambda_O \sum_{q \in Q} f_O(q_i, q_{i+1}, D) \; + \lambda_U \sum_{q \in Q} f_U(q_i, q_{i+1}, D)$$

where the potential function for unigrams is their probability estimate in Dirichlet smoothed document language model:

$$f_T(q_i, D) = \log P(q_i|\theta_D) = \log \frac{tf_{q_i,D} + \mu \frac{cf_{q_i}}{|C|}}{|D| + \mu}$$

The potential functions for ordered and unordered bigrams are defined in a similar way. SDM has 3 main parameters $(\lambda_T, \lambda_O, \lambda_U)$, which correspond to the relative contributions of potential functions for unigram, ordered bigram and unordered bigram query concepts to the final retrieval score of a document.

Previous experiments have demonstrated that taking into account term dependencies allows to significantly improve the accuracy of retrieval results compared to unigram bag-of-words retrieval models for ad-hoc document retrieval [42], particularly for longer, verbose queries [14]. The key limitation of SDM is that it considers the matches of query unigrams and bigrams in different fields of entity documents as equally important, and thus does not take into account the structure of entity documents.

5.3 Fielded Sequential and Full Dependence Models

Fielded Sequential Dependence Model (FSDM) [84], which was designed specifically for ERKG, overcomes the limitations of SDM and bag-of-words models

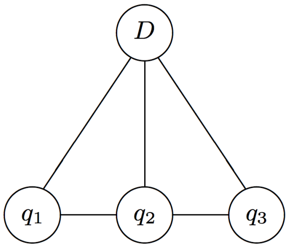


Fig. 4: **MRF graph for a 3-term query under the assumption of sequential dependencies between the query terms.**

for structured document retrieval by taking into account *both query term dependencies and document structure*. The retrieval function of FSDM quantifies the relevance of entity documents to a query *at the level of query concept types*: unigrams, ordered and unordered bigrams. In particular, each query concept type is associated with two parameters: concept type importance and the distribution of weights over the fields of entity documents. This parametrization results in the following function for scoring each structured entity document E with respect to a given query Q:

$$P_{\Lambda}(E|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, E) \; + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, E) \; + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, E)$$

where $\tilde{f}_T(q_i, E)$, $\tilde{f}_O(q_i, q_{i+1}, E)$ and $\tilde{f}_U(q_i, q_{i+1}, E)$ are the potential functions for unigrams, ordered and unordered bigrams, respectively. The potential function for unigrams in case of FSDM is defined as:

$$\tilde{f}_T(q_i, E) = \log \sum_{j=1}^{F} w_j^T P(q_i|\theta_E^j) = \log \sum_{j=1}^{F} w_j^T \frac{tf_{q_i,E^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|E^j| + \mu_j}$$

where F is the number of fields in entity document, θ_E^j is the language model of field j smoothed using its own Dirichlet prior μ_j and w_j are the field weights under the following constraints: $\sum_j w_j = 1, w_j \geq 0$; tf_{q_i,E^j} is the term frequency of q_i in field j of entity description E; $cf_{q_i}^j$ is the collection frequency of q_i in field j; $|C_j|$ is the total number of terms in field j across all entity documents in the collection and $|E^j|$ is the length of field j in E. The potential function for ordered bigrams in the retrieval function of FSDM is defined as:

$$\tilde{f}_O(q_{i,i+1}, E) = \log \sum_{j=1}^{F} w_j^O \frac{tf_{\#1(q_{i,i+1}),E^j} + \mu_j \frac{cf_{\#1(q_{i,i+1})}^j}{|C_j|}}{|E^j| + \mu_j}$$

while the potential function for unordered bigrams is defined as:

$$\tilde{f}_U(q_{i,i+1}, E) = \log \sum_{j=1}^{F} w_j^U \frac{tf_{\#uw_n(q_{i,i+1}),E^j} + \mu_j \frac{cf^j_{\#uw_n(q_{i,i+1})}}{|C_j|}}{|E^j| + \mu_j}$$

where $tf_{\#1(q_{i,i+1}),E^j}$ is the frequency of exact phrase (ordered bigram) $q_i q_{i+1}$ in field j of entity document E, $cf^j_{\#1(q_{i,i+1})}$ is the collection frequency of ordered bigram $q_i q_{i+1}$ in field j, $tf_{\#uw_n(q_{i,i+1}),E^j}$ is the number of times terms q_i and q_{i+1} co-occur within a window of n words in field j of entity document E, regardless of the order of these terms. Fielded Full Dependence Model (FFDM) is an extension of Full Dependence Model [58] to structured documents that is different from FSDM in that it takes into account all dependencies between the query terms and not just sequential ones.

In the case of structured entity documents with F fields, FSDM has a total of $3 * F + 3$ parameters (distribution of weights across F fields of entity documents for unigrams, ordered and unordered bigrams and 3 weights determining the relative contribution of potential functions for different query concept types towards the final retrieval score of an entity document). Due to its linearity with respect to the main parameters ($\boldsymbol{\lambda}$ and $\boldsymbol{w}$), the retrieval function of FSDM lends itself to efficient optimization with respect to the target retrieval metric (e.g. using coordinate ascent, which has demonstrated good performance on low-dimensional feature spaces with limited training data) [59].

Having separate mixtures of language models with different distributions of field weights for unigrams, ordered and unordered bigrams gives FSDM the flexibility to adjust the entity document scoring strategy depending on the query type. For example, the distribution of field weights, in which the matches of unordered bigrams in the descriptive fields of entity documents (attributes, categories, related entity names) have higher weights than the matches in the title fields (names, related entity names), would be more effective for informational entity queries (i.e. list search, question answering), while giving higher weights to the ordered bigram matches in the title fields would be more appropriate for navigational queries (i.e. entity search). Specifically, the accuracy and completeness of retrieval results for a list search query *"apollo astronauts who walked on the moon"* is likely to increase when more importance is given to the matches of the ordered query bigram *apollo astronauts* and unordered bigram *walked moon* in the *categories* field of entity documents, rather than in the *names* field, while giving higher weights to the matches of the same bigrams in the *name* field is likely to have the opposite effect. Experimental results [84] on publicly available benchmarks [11] indicate that additional complexity of FSDM translates into significant improvements of retrieval accuracy (20% and 52% higher MAP on entity search queries, 7% and 3% higher MAP on list search queries, 28% and 6% higher MAP on questions, 18% and 20% higher MAP on all queries) over MLM and SDM, respectively.

Hasibi et al. [39] proposed an extension of FSDM by adding a potential function that takes into account the linked entities in queries, which improves MAP by 11% on list search queries and by 16% on questions.

5.4 Parameterized Fielded Sequential and Full Dependence Models

Parametrization of entity retrieval function using distinct sets of field weights for each query concept type may still lack flexibility in some cases, which is illustrated by an example query *"capitals in Europe which were host cities of summer olympic games"*. Contrary to the assumption of FSDM, different unigrams in this query should be projected onto different fields of entity documents (i.e. "capitals" and "summer" should be projected onto the categories field, while "Europe" should be projected onto the attributes field). Mapping all these unigrams onto the same field of entity documents (either categories or attributes) is likely to degrade the accuracy of retrieval results for this query.

Parameterized Fielded Sequential Dependence Model (PFSDM) [65] is an extension of FSDM that provides a more flexible parametrization of entity retrieval function by estimating the importance weight for matches of *each individual query concept* (unigram or bigram), rather than each *query concept type*, in different fields of entity documents. Specifically, PFSDM uses the same potential functions as FSDM, but estimates $w^T_{q_i,j}$, the relative contribution of each individual query unigram q_i, and $w^{O,U}_{q_{i,i+1},j}$, the relative contribution of each individual query bigram $q_{i,i+1}$ (ordered or unordered), which are matched in field j of structured entity document for entity E towards the retrieval score of E, as a linear combination of features:

$$w^T_{q_i,j} = \sum_k \alpha^U_{j,k}\phi_k(q_i, j)$$

$$w^{O,U}_{q_{i,i+1},j} = \sum_k \alpha^B_{j,k}\phi_k(q_{i,i+1}, j)$$

under the following constraints:

$$\sum_j w^T_{q_i,j} = 1, w^T_{q_i,j} \geq 0, \alpha^U_{j,k} \geq 0, 0 \leq \phi_k(q_i, j) \leq 1$$

$$\sum_j w^{O,U}_{q_{i,i+1},j} = 1, w^{O,U}_{q_{i,i+1},j} \geq 0, \alpha^B_{j,k} \geq 0, 0 \leq \phi_k(q_{i,i+1}, j) \leq 1$$

where $\phi_k(q_i, j)$ and $\phi_k(q_{i,i+1}, j)$ are the values of the k-th non-negative feature function for query unigram q_i and bigram $q_{i,i+1}$ in field j of entity document, respectively. $w^T_{q_i,j}$ and $w^{O,U}_{q_{i,i+1},j}$ can also be considered as a dynamic projection of query unigrams q_i and bigrams $q_{i,i+1}$ onto the fields of structured entity documents. Similar to FFDM, Parameterized Fielded Full Dependence Model (PFFDM) takes into account all dependencies between the query terms and not

Table 1: **Features to estimate the contribution of query concept κ matched in field j towards the retrieval score of E. Column CT designates the type of query concept that a feature is used for (UG stands for unigrams, BG stands for bigrams)**.

Source	**Feature**	**Description**	**CT**
Collection statistics	$FP(\kappa, j)$	Posterior probability $P(E_j\|w)$ obtained through Bayesian inversion of $P(w\|E_j)$, as defined in [46].	UG BG
	$TS(\kappa, j)$	Retrieval score of the top document according to SDM [58], when concept κ is used as a query and only the jth fields of entity representations are used as documents.	BG
Stanford POS Tagger[5]	$NNP(\kappa)$	Is concept κ a proper noun (singular or plural)?	UG
	$NNS(\kappa)$	Is concept κ a plural non-proper noun? We consider a bigram as plural when at least one of its terms is plural.	UG BG
	$JJS(\kappa)$	Is concept κ a superlative adjective?	UG
Stanford Parser[6]	$NPP(\kappa)$	Is concept κ part of a noun phrase?	BG
	$NNO(\kappa)$	Is concept κ the only singular non-proper noun in a noun phrase?	UG
	INT	Intercept feature, which has value 1 for all concepts.	UG BG

just sequential ones. The features that are used by PFSDM and PFFDM to estimate the projection of a query concept κ onto the field j of structured entity document are summarized in Table 1.

As follows from Table 1, PFSDM uses two types of features: real-valued features (FP, TS), which are based on the collection statistics of query concepts in a particular field of entity documents, and binary features (NNP, NNS, JJS, NPP, NNO), which are based on the output of natural language processing tools (POS tagger and syntactic parser) and are independent of the fields of entity documents. The intuition behind the latter type of features is that the relationship between them and the fields of entity documents can be learned in the process of estimating their weights. For example, since plural non-proper nouns typically indicate groups of entities, the weight of the corresponding feature (NNS) is likely to be higher in the *categories* field than in all other fields of entity documents. On the other hand, the NNP feature takes positive values for the query concepts that are proper nouns and designate a specific entity. Therefore, the distribution of field weights for this feature is likely to be skewed towards *names*, *similar entity names* and *related entity names* fields. Unlike PRMS [46], PFSDM and PFFDM estimate the projections of query concepts onto the fields of entity documents based on multiple features of different type, which allows to overcome the issue of sparsity for entity representations with large number of fields and increase the robustness of estimates of these projections. In the case of structured entity documents with F fields, PFSDM and PFFDM have $F * U + F * B + 3$ parameters in total ($F * U$ feature weights for unigrams and $F * B$ feature weights for bigrams, where U and B are the number of features for unigrams and bigrams, and 3 weights determining the relative contribution of potential functions for each query concept type towards the final retrieval score of entity document). Similar to FSDM and FFDM, feature weights can be optimized with respect to the target retrieval metric using any derivative-free optimization method (e.g. coordinate ascent). Experimental results [65] on publicly available benchmarks [11] indicate that more flexible parametrization of entity relevance and feature-based estimation of field mapping weights by PFSDM yields significant improvements of retrieval accuracy (87% and 7% higher MAP on entity search queries, 82% and 12% higher MAP on questions, 77% and 4% higher MAP on all queries) over PRMS and FSDM, respectively.

6 Entity set expansion

An initial set of entities retrieved for a given keyword query or a question in the first stage of entity retrieval process using BM25 [76], BM25F [15, 16, 32, 76], Kullback-Leibler divergence [8, 9, 34], Mixture of Language Models (MLM) [19, 61, 64, 83], FSDM/FFDM or PFSDM/PFFDM can be expanded in the second stage with additional entities and entity attributes obtained using the methods based on SPARQL queries and spreading activation.

6.1 SPARQL queries

Tonon et al. [76] proposed a hybrid entity retrieval and expansion method that maintains an inverted index for entity documents and a triple store for entity relations. The method first retrieves an initial set of entities from the inverted index of flat (non-structured) entity documents using BM25 retrieval model and expands the initial set of entities with their attributes, neighbor entities and neighbors of neighbor entities found by issuing pre-defined SPARQL queries to the triple store. Besides general predicates, such as `owl:sameAs` and `skos:subject`, SPARQL queries mostly leverage DBpedia specific predicates, such as `dbpedia:wikilink`, `dbpedia:disambiguates` and `dbpedia:redirect`. Expansion entities are evaluated with respect to the original query using Jaro-Winkler similarity score and the entities, for which the similarity score is below a given threshold, are filtered out. Original and expansion entities are then reranked based on a linear combination of BM25 and Jaro-Winkler scores. Experiments indicate that, for entity search queries, expansion of the original entity set retrieved using BM25 by following just `owl:sameAs` predicates results in 9-11% increase in MAP. Following `dbpedia:redirect` and `dbpedia:disambiguates` predicates, in addition to `owl:sameAs`, results in 12-25% increase in MAP. However, following other general predicates (`dbpedia:wikilink`, `skos:subject`, `foaf:homepage`, etc.) and looking further into a KG (i.e. expanding with neighbors of neighbor entities) degrades the initial retrieval results (similar findings were reported in [6, 48] for term graphs and semantic networks).

6.2 Spreading activation

A general approach based on weighted spreading activation on KGs to expand the initial set of entities obtained using any retrieval model was proposed in [72]. The SemSets method [22] proposed for list search utilizes the relevance of entities to automatically constructed categories (i.e. semantic sets) measured according to structural and textual similarity. This approach combines a retrieval model (basic TF-IDF retrieval model) with the ranking method based on spreading activation over the link structure of a knowledge graph to evaluate the membership of entities in semantic sets.

7 Entity ranking

Ranking the expanded set of entities is the final stage in ERKG pipeline. In this section, we provide an overview of recent research on transfer learning, incorporating latent semantics and ranking entities in document search results.

7.1 Transfer learning

Dali and Fortuna [24] manually converted keyword queries into SPARQL queries and examined the utility of machine learning methods for ranking the retrieved entities using ranking SVM. In particular, they used the following types of features capturing the popularity and importance of entity E:

- **Wikipedia popularity features**: popularity of E measured by the statistics of the Wikipedia page for E, such as page length, the number of page edits and the number of page accesses from Wikipedia logs;
- **Search engine popularity features**: popularity of E measured by the number of results returned by a search engine API using the top 5 keywords from the abstract of the Wikipedia page for E as a query;
- **Web popularity features**: number of occurrences of entity name in Google N-grams;
- **KG importance features**: importance of E measured by the number of triples, in which E is a subject (i.e. entity node out-degree); the number of triples, in which E is an object (i.e. entity node in-degree); the number of triples, in which E is a subject and object is a literal as well as the number of categories and the sizes of the biggest, smallest, median category that the Wikipedia page for E belongs to;
- **KG centrality features**: HITS hub and authority scores and Pagerank of both the Wikipedia page for E in Wikipedia graph and of entity node in a KG.

Two experiments were performed using these features. The first experiment focused on studying the effectiveness of individual features and led to several interesting conclusions. First, features approximating entity importance as HITS scores of Wikipedia page corresponding to an entity in Wikipedia graph are effective for entity ranking, while PageRank and HITS scores of entity nodes in a knowledge graph are not. Second, Google N-grams are a cheaper proxy for search engine API in determining entity popularity. The second experiment was aimed at assessing the feasibility of transfer learning for entity ranking. Specifically, the ranking model was first trained on DBpedia entities and then applied to rank YAGO entities. The results of this experiment indicate that, in general, ranking models for different knowledge graphs are non-transferable, unless they involve a large number of features. The largest drops of performance were observed when the ranking model was trained on KG-specific features, which suggests that different KGs have their own peculiarities reflecting the decisions of their creators, which are non-generalizable.

7.2 Leveraging Latent Semantics in Entity Ranking

Numerous approaches [17, 53, 52, 78] to model latent semantics of entities in KGs have been proposed in recent years. RESCAL [63], a tensor factorization-based method for relational learning, obtains low-dimensional entity representations by factorizing a sparse tensor $\mathcal{X}$ of size $n \times n \times m$, where n is the number of distinct entities and m is the number of distinct predicates in a KG. Binary tensor $\mathcal{X}$ is constructed in such a way that each of its frontal slices corresponds to a sparse adjacency matrix of a subgraph of a KG involving a particular predicate. If entities i and j are connected with predicate k in a KG, then $\mathcal{X}_{ijk} = 1$, otherwise $\mathcal{X}_{ijk} = 0$.

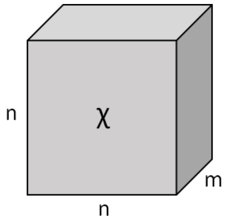


Fig. 5: **Representation of a KG as a binary tensor. Each frontal slice corresponds to an adjacency matrix of a subgraph of a KG involving a particular predicate.**

RESCAL factorizes $\mathcal{X}$ in such a way that each frontal slice X_k is approximated with a product of three matrices:

$$X_k \approx AR_kA^T, \text{ for } k = 1, \ldots, m$$

where A is a $n \times r$ matrix, in which the ith row corresponds to an r-dimensional latent representation (i.e. embedding) of the ith entity in a KG (r is specified by a user) and R is an interaction tensor, in which each frontal slice R_k is a dense $r \times r$ square matrix that models the interactions of latent components of entity representation the k-th predicate. Figure 6 shows a graphical representation of such factorization.

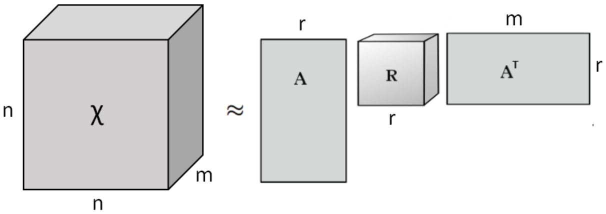


Fig. 6: **Graphical representation of knowledge graph tensor factorization using RESCAL.**

A and R_k are computed by solving the following optimization problem:

$$\min_{A,R} \frac{1}{2}\left(\sum_k \|X_k - AR_kA^T\|_F^2\right) + \lambda\left(\|A\|_F^2 + \sum_k \|R_k\|_F^2\right)$$

using an iterative alternating least squares algorithm.

Zhiltsov and Agichtein [83] utilized KG entity embeddings obtained using RESCAL to derive structural entity similarity features that were used in a machine learning method for ranking the results of entity retrieval models. Specifically, their approach re-ranked the retrieval results of MLM using Gradient Boosted Regression Trees in conjunction with term-based and structural features. Term-based features include query length and query clarity, entity retrieval score using MLM with uniform field weights as well as bigram relevance scores for each of the fields in 3-field entity document. Structural features are based on distance metrics in the latent space between embedding of a given entity and embeddings of the top-3 entities retrieved by the baseline method (MLM). Experiments indicate that a combination of term-based and structural features improves MAP, NDCG and P@10 by 5-10% relative to MLM on entity search queries.

7.3 Ranking entities in search results

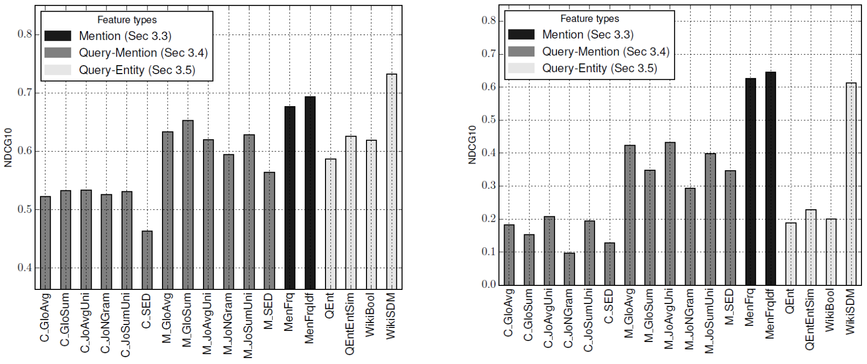
An alternative method to retrieving and ranking entities directly from a KG was proposed by Schuhmacher et al. [73]. Their method is based on linking entity mentions in top retrieved documents to KG entities and ranking the linked

Table 2: **Features for ranking entities linked to entity mentions in retrieved documents**.

Mention Features	
MenFrq	number of entity occurrences in top documents
MenFrqIdf	IDF of entity mention
Query-Mention Features	
SED	normalized Levenshtein distance
Glo	similarity based on GloVe embeddings
Jo	similarity based on JoBimText embeddings
Query-Entity Features	
QEnt	is document entity linked in query
QEntEntSim	is there a path in KG between document and query entities
WikiBoolean	is entity Wikipedia article retrieved by query using Boolean model
WikiSDM	SDM retrieval score of entity Wikipedia article using query
Entity-Entity Features	
Wikipedia	is there a path between two entities in DBpedia KG

entities using ranking SVM in conjunction with mention, query-mention, query-entity and entity-entity features summarized in Table 2.

Using this method, entities can be retrieved and ranked for any free-text Web-style queries (e.g. *"Argentine British relations"*), which aim at heterogeneous entities with no specific target type, and presented next to traditional document results.



a) ClueWeb12 **b) Robust04**

Fig. 8: **Ranking performance of each feature on different collections.**

Analysis of ranking performance of each individual feature (summarized in Figure 8) resulted in several interesting conclusion. First, the strongest features are the IDF of entity mention (MenFrqIdf) and SDM retrieval score of entity

Wikipedia page (WikiSDM). Second, all context-based query-mention features (indicated by prefix C_) perform worse than their non-context counterparts (indicated by prefix M_). Third, other query-entity features based on DBpedia (QEnt and QEntEntSim) perform worse than WikiSDM, but better than other mention-based features. In addition to these finding, feature ablation studies revealed that DBpedia-based features have positive, but insignificant influence on performance, while Wikipedia-based features show strong and significant influence. Furthermore, authoritativeness of entities marginally correlates with their relevance, since entities that have high PageRank scores are typically very general and are linked to by many other entities.

8 Conclusion

The past decade has witnessed the emergence of numerous large-scale publicly available (e.g. DBpedia, Wikidata and YAGO) and proprietary (e.g. Google's Knowledge Graph, Facebook's Open Graph and Microsoft's Satori) knowledge graphs. However, we only begin to understand how to effectively access and utilize vast amounts of information stored in them. This tutorial is an attempt to summarize and systematize the published research related to accessing information in knowledge graphs. Specific goals of this tutorial are two-fold. On one hand, we outlined a typical architecture of systems for searching entities in knowledge graphs and reported the best practices known for each component of those systems, in order to facilitate their rapid development by practitioners. On the other hand, we summarized the recent advances and main ideas related to entity representation, retrieval and ranking as well as entity set expansion with an intent of helping information retrieval and machine learning researchers to initiate their own research into these directions and produce exciting discoveries in many years to come.

References

1. B. Aditya, Gaurav Bhalotia, Soumen Chakrabarti, Arvind Hulgeri, Charuta Nakhe, Parag Parag, and S. Sundarsan. BANKS: Browsing and Keyword Searching in Relational Databases. In *Proceedings of the 28th International Conference on Very Large Databases*, pages 1083–1086, 2002.
2. Sihem Amer-Yahia, Nick Koudas, Amelie Marian, Divesh Srivastava, and David Toman. Structure and Content Scoring for XML. In *Proceedings of the 31st International Conference on Very Large Databases*, pages 361–372, 2005.
3. Rajul Anand and Alexander Kotov. Improving difficult queries by leveraging clusters in term graph. In *Proceedings of the 11th Asia Information Retrieval Symposium*, pages 426–432, 2015.
4. Saeid Balaneshinkordan and Alexander Kotov. An empirical comparison of term association and knowledge graphs for query expansion. In *Proceedings of the 38th European Conference on Information Retrieval Research*, pages 761–767, 2016.
5. Saeid Balaneshinkordan and Alexander Kotov. Optimization method for weighting explicit and latent concepts in clinical decision support queries. In *Proceedings of*

the 2nd ACM International Conference on the Theory of Information Retrieval, pages 241–250, 2016.

6. Saeid Balaneshinkordan and Alexander Kotov. Sequential query expansion using concept graph. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 155–164, 2016.
7. Krisztian Balog, Leif Azzopardi, and Maarten 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, 2006.
8. Krisztian Balog, Marc Bron, and Maarten de Rijke. Category-based Query Modeling for Entity Search. In *Proceedings of the 32nd European Conference on Information Retrieval*, pages 319–331, 2010.
9. Krisztian Balog, Marc Bron, and Maarten de Rijke. Query Modeling for Entity Search based on Terms, Categories, and Examples. *ACM Transactions on Information Systems*, 29(22), 2011.
10. Krisztian Balog, Arjen P. de Vries, Pavel Serdyukov, Paul Thomas, and Thijs Westerveld. Overview of the TREC 2009 Entity Track. In *Proceedings of the 18th Text REtrieval Conference*, 2010.
11. Krisztian Balog and Robert Neumayer. A Test Collection for Entity Search in DBpedia. In *Proceedings of the 36th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 737–740, 2013.
12. Krisztian Balog, Pavel Serdyukov, and Arjen P. de Vries. Overview of the TREC 2011 Entity Track. In *Proceedings of the 20th Text REtrieval Conference*, 2012.
13. Krisztian Balog, Wouter Weerkamp, and Maarten De Rijke. A few examples go a long way: constructing query models from elaborate query formulations. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 371–378, 2008.
14. Michael Bendersky, Donald Metzler, and W. Bruce Croft. Learning Concept Importance Using a Weighted Dependence Model. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 31–40, 2010.
15. Roi Blanco, Harry Halpin, Daniel M. Herzig, Peter Mika, Jeffrey Pound, and Henry S. Thompson. Entity Search Evaluation over Structured Web Data. In *Workshop on Entity Oriented Search*, 2011.
16. Roi Blanco, Peter Mika, and Sebastiano Vigna. Effective and Efficient Entity Search in RDF Data. In *Proceedings of the 10th International Conference on the Semantic Web*, pages 83–97, 2011.
17. Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the Neural Information Processing Systems*, pages 2787–2795, 2013.
18. Marc Bron, Krisztian Balog, and Maarten de Rijke. Ranking Related Entities: Components and Analyses. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1079–1088, 2010.
19. Marc Bron, Krisztian Balog, and Maarten de Rijke. Example Based Entity Search in the Web of Data. In *Proceedings of the 35th European Conference on Information Retrieval*, pages 392–403, 2013.
20. Surajit Chaudhuri, Gautam Das, Vagelis Hristidis, and Gerhard Weikum. Probabilistic Information Retrieval Approach for Ranking of Database Query Results. *ACM Transactions on Database Systems*, 31:1134–1168, 2006.
21. Tao Cheng, Xifeng Yan, and Kevin Chen-Chuan Chang. EntityRank: Searching Entities Directly and Holistically. In *Proceedings of the 33rd International Conference on Very Large Databases*, pages 387–398, 2007.

22. Marek Ciglan, Kjetil Nørvåg, and Ladislav Hluchý. The SemSets Model for Adhoc Semantic List Search. In *Proceedings of the 21st World Wide Web Conference*, pages 131–140, 2012.
23. William W. Cohen. Integration of Heterogeneous Databases without Common Domains using Queries based on Textual Similarity. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 201–212, 1998.
24. Lorand Dali and Blaž Fortuna. Learning to rank for semantic search. In *Proceedings of the 4th International Semantic Search Workshop*, 2011.
25. Jeffrey Dalton, Laura Dietz, and James Allan. Entity Query Feature Expansion Using Knowledge Base Links. In *Proceedings of the 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 365–374, 2014.
26. Arjen P. de Vries, Anne-Marie Vercoustre, James A. Thom, Nick Craswell, and Mounia Lalmas. Overview of the INEX 2007 Entity Ranking Track. *Lecture Notes in Computer Science*, 4862:245–251, 2008.
27. Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Neijdl. Why Finding Entities in Wikipedia is Difficult Sometimes. *Information Retrieval*, 13:534–567, 2010.
28. Gianluca Demartini, Tereza Iofciu, and Arjen P. de Vries. Overview of the INEX 2009 Entity Ranking Track. In *Proceedings of INEX'09*, 2009.
29. Gianluca Demartini, Tereza Iofciu, and Arjen P. de Vries. Overview of the INEX 2009 Entity Ranking Track. *Lecture Notes in Computer Science*, 6203:254–264, 2010.
30. Shady Elbassuoni and Roi Blanco. Keyword Search over RDF Graphs. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, pages 237–242, 2011.
31. Shady Elbassuoni, Maya Ramanath, Ralf Schenkel, Marcin Sydow, and Gerhard Weikum. Language-model-based Ranking for Queries on RDF-graphs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 977–986, 2009.
32. Besnik Fetahu, Ujwal Gadiraju, and Stefan Dietze. Improving Entity Retrieval on Structured Data. In *Proceedings of the 14th International Semantic Web Conference*, pages 474–491, 2015.
33. Konstantin Golenberg, Benny Kimelfeld, and Yehoshua Sagiv. Keyword Proximity Search in Complex Data Graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 927–940, 2008.
34. Swapna Gottipati and Jing Jiang. Linking Entities to a Knowledge Base with Query Expansion. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 804–813, 2011.
35. David Graus, Manos Tsagkias, Wouter Weerkamp, Edgar Meij, and Maarten de Rijke. Dynamic collective entity representations for entity ranking. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pages 595–604, 2016.
36. R. Guha, Rob McCool, and Eric Miller. Semantic Search. In *Proceedings of the 12th International Conference on World Wide Web*, pages 700–709, 2003.
37. Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named Entity Recognition in Query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–274, 2009.

38. Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Entity Linking in Queries: Tasks and Evaluation. In *Proceedings of the 1st ACM International Conference on the Theory of Information Retrieval*, pages 171–180, 2015.
39. Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2nd ACM International Conference on the Theory of Information Retrieval*, pages 209–218, 2016.
40. Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.
41. Vagelis Hristidis, Heasoo Hwang, and Yannis Papakonstantinou. Authority-based Keyword Search in Databases. *ACM Transactions on Database Systems*, 13(1), 2008.
42. Samuel Huston and W. Bruce Croft. A Comparison of Retrieval Models using Term Dependencies. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 111–120, 2014.
43. Varun Kacholia, Shashank Pandit, Soumen Chakrabarti, S. Sudarshan, Rushi Desai, and Hrishikesh Karambelkar. Bidirectional Expansion for Keyword Search on Graph Databases. In *Proceedings of the 31st International Conference on Very Large Databases*, pages 505–516, 2005.
44. Rianne Kaptein and Jaap Kamps. Exploiting the Category Structure of Wikipedia for Entity Ranking. *Artificial Intelligence*, 194:111–129, 2013.
45. Rianne Kaptein, Pavel Serdyukov, Arjen de Vries, and Jaap Kamps. Entity Ranking using Wikipedia as a Pivot. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 69–78, 2010.
46. Jin Young Kim, Xiaobing Xue, and W. Bruce Croft. A Probabilistic Retrieval Model for Semistructured Data. In *Proceedings of the 31st European Conference on Information Retrieval*, pages 228–239, 2009.
47. Alexander Kotov and ChengXiang Zhai. Interactive sense feedback for difficult queries. In *Proceedings of 20th ACM International Conference on Information and Knowledge Management*, pages 163–172, 2011.
48. Alexander Kotov and ChengXiang Zhai. Tapping into knowledge base for concept feedback: Leveraging conceptnet to improve search results for difficult queries. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 403–412, 2012.
49. Joonseok Lee, Ariel Fuxman, Bo Zhao, and Yuanhua Lv. Leveraging Knowledge Bases for Contextual Entity Exploration. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1949–1958, 2015.
50. Guoliang Li, Beng Chin Ooi, Jianhua Feng, Jianyong Wang, and Lizhu Zhou. EASE: an Effective 3-in-1 Keyword Search Method for Unstructured, Semi-structured and Structured Data. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 903–914, 2008.
51. Thomas Lin, Patrick Pantel, Michael Gamon, Anitha Kannan, and Ariel Fuxman. Active Objects Actions for Entity Centric Search. In *Proceedings of the 21st International Conference on World Wide Web*, pages 589–598, 2012.
52. Yankai Lin, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning with entities, attributes and relations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2866–2872, 2016.
53. Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2181–2187, 2015.

54. Fang Liu, Clement Yu, Weiyi Meng, and Abdur Chowdhury. Effective Keyword Search in Relational Databases. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pages 563–574, 2006.
55. Hugo Liu and Push Singh. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.
56. Xitong Liu and Hui Fang. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*, 18(6):473–503, 2015.
57. Xitong Liu, Wei Zheng, and Hui Fang. An Exploration of Ranking Models and Feedback Method for Related Entity Finding. *Information Processing and Management*, 49:995–1007, 2013.
58. Donald Metzler and W. Bruce Croft. A Markov Random Field Model for Term Dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 472–479, 2005.
59. Donald Metzler and W. Bruce Croft. Linear Feature-based Models for Information Retrieval. *Information Retrieval*, 10:257–274, 2007.
60. Iris Miliaraki, Roi Blanco, and Mounia Lalmas. From “Selena Gomez” to “Marlon Brando”: Understanding Explorative Entity Search. In *Proceedings of the 24th International Conference on World Wide Web*, pages 765–775, 2015.
61. Robert Neumayer, Krisztian Balog, and Kjetil Nørvåg. On the Modeling of Entities for Ad-hoc Entity Search in the Web of Data. In *Proceedings of the 34th European Conference on Information Retrieval*, pages 133–145, 2012.
62. Robert Neumayer, Krisztian Balog, and Kjetil Nørvåg. When Simple is (more than) Good Enough: Effective Semantic Search with (almost) no Semantics. In *Proceedings of the 34th European Conference on Information Retrieval*, pages 540–543, 2012.
63. Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280, 2012.
64. Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, and Wei-Ying Ma. Web Object Retrieval. In *Proceedings of the 16th International Conference on World Wide Web*, pages 81–90, 2007.
65. Fedor Nikolaev, Alexander Kotov, and Nikita Zhiltsov. Parameterized fielded term dependence models for ad-hoc entity retrieval from knowledge graph. In *Proceedings of the 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 435–444, 2016.
66. Paul Ogilvie and Jamie Callan. Combining Document Representations for Known-item Search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 143–150, 2003.
67. José R. Pérez-Aguera, Javier Arroyo, Jane Greenberg, Joaquin Perez Iglesias, and Victor Fresno. Using BM25F for Semantic Search. In *Proceedings of the 3rd International SemSearch Workshop*, 2010.
68. Jay M Ponte and W Bruce 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, 1998.
69. Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc Object Retrieval in the Web of Data. In *Proceedings of the 19th World Wide Web Conference*, pages 771–780, 2010.

70. Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.
71. Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pages 42–49, 2004.
72. Cristiano Rocha, Daniel Schwabe, and Marcus Poggi de Argão. A Hybrid Approach for Searching in the Semantic Web. In *Proceedings of the 13th International Conference on World Wide Web*, pages 374–383, 2004.
73. Michael Schuhmacher, Laura Dietz, and Simone Paolo Ponzetto. Ranking Entities for Web Queries Through Text and Knowledge. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pages 1461–1470, 2015.
74. Michael Schuhmacher and Simone Paolo Ponzetto. Knowledge-based Graph Document Modeling. In *Proceedings of the 7th ACM ACM International Conference on Web Search and Data Mining*, pages 543–552, 2014.
75. Saeedeh Shakarpour, Axel-Cyrille Ngonga Ngomo, and Sören Auer. Question Answering on Interlinked Data. In *Proceedings of the 22nd World Wide Web Conference*, pages 1145–1156, 2013.
76. Alberto Tonon, Gianluca Demartini, and Philippe Cudré-Mauroux. Combining Inverted Indices and Structured Search for Ad-hoc Object Retrieval. In *Proceedings of the 35th International ACM Conference on Research and Development in Information Retrieval*, pages 125–134, 2012.
77. Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based Question Answering over RDF data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 639–648, 2012.
78. Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.
79. Chenyan Xiong and Jamie Callan. EsdRank: Connecting Query and Documents through External Semi-Structured Data. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, pages 951–960, 2015.
80. Chenyan Xiong and Jamie Callan. Query Expansion with Freebase. In *Proceedings of the 2015 ACM International Conference on The Theory of Information Retrieval*, pages 111–120, 2015.
81. Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, and Gerhard Weikum. Robust Question Answering over the Web of Linked Data. In *Proceedings of the 22nd ACM Conference on Information and Knowledge Management*, pages 1107–1116, 2013.
82. Xifeng Yan, Philip S. Yu, and Jiawei Han. Substructure Similarity Search in Graph Databases. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pages 766–777, 2006.
83. Nikita Zhiltsov and Eugene Agichtein. Improving Entity Search over Linked Data by Modeling Latent Semantics. In *Proceedings of the 22nd ACM Conference on Information and Knowledge Management*, pages 1253–1256, 2013.
84. Nikita Zhiltsov, Alexander Kotov, and Fedor Nikolaev. Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data. In *Proceedings of the 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–262, 2015.