Noname manuscript No.
(will be inserted by the editor)

A Review of Graph-Based Models for Entity-Oriented Search

José Devezas* · Sérgio Nunes

the date of receipt and acceptance should be inserted later

Abstract Entity-oriented search tasks heavily rely on exploiting unstructured and structured collections. Moreover, it is frequent for text corpora and knowledge bases to provide complementary views on a common topic. While, traditionally, the retrieval unit was the document, modern search engines have evolved to also retrieve entities and to provide direct answers to the information needs of the users. Cross-referencing information from heterogeneous sources has become fundamental, however a mismatch still exists between text-based and knowledge-based retrieval approaches. The former does not account for complex relations, while the latter does not properly support keyword-based queries and ranked retrieval. Graphs are a good solution to this problem, since they can be used to represent text, entities and their relations. In this survey, we examine text-based approaches and how they evolved in order to leverage entities and their relations in the retrieval process. We also cover multiple aspects of graph-based models for entity-oriented search, providing an overview on link analysis and exploring graph-based text representation and re-

José Devezas is supported by research grant PD/BD/128160/2016, provided by the Portuguese national funding agency for science, research and technology, Fundação para a Ciência e a Tecnologia (FCT), within the scope of Operational Program Human Capital (POCH), supported by the European Social Fund and by national funds from MCTES.

J. Devezas*
INESC TEC and Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
Tel.: +351-225-082-134
Fax: +351-225-574-103
E-mail: jld@fe.up.pt
ORCID: 0000-0003-2780-2719
* Corresponding author

S. Nunes
INESC TEC and Faculty of Engineering, University of Porto
Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
Tel.: +351-225-082-134
Fax: +351-225-574-103
ORCID: 0000-0002-2693-988X
E-mail: ssn@fe.up.pt

trieval, leveraging knowledge graphs for document or entity retrieval, building entity graphs from text, using graph matching for querying with subgraphs, exploiting hypergraph-based representations, and ranking based on random walks on graphs. We close with a discussion on the topic and a view of the future to motivate the research of graph-based models for entity-oriented search, particularly as joint representation models for the generalization of retrieval tasks.

Keywords Entity-oriented search · Graph-based models · Hypergraph-based models · Random walk based models

1 Introduction

In 1990, Alan Emtage [50] created Archie[1], the first internet search engine, built to locate content on public FTP servers. At that time, search was still heavily based on keyword queries, as inspired by the library and the search potential of the back-of-the-book index. However, with the evolution of the web and the devices used to interact with it, the materialization of people's information needs also evolved. Queries changed from simple topic-driven keywords to more complex entity-oriented structures. In 2007, Bautin and Skiena [21] found that nearly 87% of all queries contained entities, according to the analysis of 36 million queries released by AOL [7]. Furthermore, entities are also frequently found in documents — in the CoNLL 2003 English training set [128], there are 1.6 entities per sentence (23,499 entities for 14,987 sentences). Such a pervasive presence of entities, both in queries and in documents, easily justifies the current direction of search engines and their focus on entity-oriented search.

According to Balog [12][Def.1.5]:

> *Entity-oriented search is the search paradigm of organizing and accessing information centered around entities, and their attributes and relationships.*

This clashes with the classical definition of information retrieval as portrayed by Manning et al. [98]:

> *Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).*

In entity-oriented search, the materials can be of an unstructured or structured nature. In fact, they are often a combination of both, either taking the form of semi-structured data or links between unstructured and structured data. In their survey on semantic search on text and knowledge bases, Bast et al. [19][Def.2.3] defined combined data as text annotated with entities from a knowledge base, or as a combination of knowledge bases with different naming schemes. Combined data is at the core of entity-oriented search. However, in the past, techniques for representing and querying corpora and knowledge bases have been explored separately. In a way, two different communities are now intersecting. Appropriately, Baeza-Yates et al. [10] had identified semantic search as a task that lies in between several areas

[1] `http://archie.icm.edu.pl/archie-adv_eng.html`

of specialization. The same applies to entity-oriented search, which, according to Balog [12][§1.3.3], is subsumed by semantic search.

Modern search engines accomplish entity-oriented search through the orchestration of several components which are built on top of a common set of resources — a collection of documents and/or knowledge bases, containing terms and entities, along with links and resource statistics. A complete pipeline relies on components for entity ranking and similarity measurement, target entity type identification, word sense disambiguation and entity linking, document and query semantic analysis, query expansion and entity list completion, and query recommendation and related entity finding. Some of these approaches can be unified and, while not exclusively limited to, this can often be accomplished through graph-based approaches. Take for instance Moro et al. [104] who proposed a graph-based approach for unified word sense disambiguation and entity linking, Ganea and Hoffman [62], who developed a joint representation of words and entities through vector space embeddings, or even Richardson and Domingos [124], who proposed Markov logic networks, as a combination of probability and first-order logic, which could easily model the uncertainty of statements describing entity attributes and relationships.

The remainder of this document is organized as follows:

- **Section 2** covers the motivation behind this survey, built on top of the needs to represent combined data and improve the effectiveness of retrieval tasks in entity-oriented search.
- **Section 3** describes the literature review methodology that we used, providing basic statistics about the considered publications.
- **Section 4** introduces the classical models of information retrieval and how they influenced and led to applications in entity-oriented search [§4.1]. It also introduces learning to rank, highlighting entity-oriented search applications [§4.2].
- **Section 5** focuses on describing graph-based models with strategies applicable to entity-oriented search. We start by introducing classical link analysis [§5.1], and text representations as a graph [§5.2]. We cover retrieval processes based on knowledge graphs, as well as their construction [§5.3]. We then study retrieval strategies based on entity graphs directly built from text [§5.4], and explore their tensor-based representation [§5.5]. We also cover graph matching, which is an important part of the semantic web, used in SPARQL for querying RDF [§5.6]. We cover several hypergraph-based models, used for different representation and retrieval tasks, including unified indexes, modeling complex document structures, or establishing higher-order dependencies to rank documents [§5.7]. Finally, we survey random walk based models, focusing on PageRank variations with several concrete applications in entity-oriented search [§5.8].
- **Section 6** begins by presenting several observations about the area [§6.1]: justifying the need for a state of the art in graph-based entity-oriented search; commenting on the relations between entity-oriented search and semantic search; clarifying the definition of graph-based models, as used throughout this survey and across the literature. We then provide an overview on the reviewed strategies for entity-oriented search [§6.2], segmenting them by approach and tasks, for classical, learning to rank, and graph-based models.
- **Section 7** closes with final remarks and a reflection on the future of graph-based entity-oriented search.

2 Motivation

Entity-oriented search not only encompasses tasks based on entity ranking, such as ad hoc entity retrieval, related entity finding, and entity list completion, but it also covers ad hoc document retrieval, as long as it relies on entities for semantic enrichment [12][Ch.8]. While these tasks can be modeled individually, they share a common collection of combined data, bringing together text and entities, in their heterogeneity, through annotations that connect mentions to entities, as well as individuals representing the same entity. A data structure capable of representing such heterogeneous data is a graph, which is why this survey focuses on exploring graph-based entity-oriented search. Graphs have the ability to represent documents, entities, and their relations, working as a joint representation model and presenting the opportunity to approach general information retrieval.

Our goal with this survey is to provide an overview of the available graph-based mechanisms that can be used to innovate and support the joint representation of corpora and knowledge bases, in order to build universal ranking functions across multiple retrieval tasks.

We first look into classical retrieval and learning to rank models, along with their specific applications in entity-oriented search, so that we can understand which tasks are being researched, how they are being tackled and, when available, how generalization is approached. This review stage is akin to requirements elicitation in engineering design process, enabling us to establish the needs and features for a good graph-based model for entity-oriented search.

We then move into graph-based models and approaches that can be useful for entity-oriented search. The goal of this review stage is to compile and categorize useful literature that illustrates a wide range of applications and mechanisms that can be integrated into a general model for entity-oriented search. This includes approaches for:

- Querying, ranking, or defining weighting schemes based on a graph (graph matching, link analysis, random walk based models, etc.);
- Representing text as a graph (e.g., based on the relationships between terms, or among documents);
- Representing entities as a graph, be it through manually curated knowledge graphs, or based on entity graphs automatically generated from text;
- Generalizing models for representation and retrieval of text and entities (e.g., tensor-based graph representations, hypergraph-based models, etc.);
- Evaluating entity-oriented search tasks, including evaluation forums and test collections.

Our motivation was to gather and organize otherwise scattered literature in a way that would be useful for innovating around graph-based entity-oriented search, focusing on unifying models, both for providing joint representations of corpora and knowledge bases, and to further motivate the research of universal or general ranking approaches.

3 Methodology

We relied on an exploratory literature review approach, refining and refocusing along the process, as concepts became clearer, over a period of five years. We used aca-

demic search engines to issue queries in an attempt to solve our information needs about approaches that could be useful for the representation and retrieval of corpora and knowledge bases using graph-based models. Resulting publications were selected by reading the title, the abstract, the conclusions, and sometimes a part of the introduction, in this order.

Through this approach, we were able to identify 203 publications, written by 492 distinct authors, ranging from 1950 to 2021. The considered literature also covered 81 conferences, with a CORE rank ranging from A* to C, as well as 47 journals, with a SCImago Journal Rank indicator ranging from 0.178 to 6.08, and a journal h-index ranging from 22 to 699. The collected literature is surveyed in the sections that follow.

4 From text-based to entity-oriented search

Until recently, search has been focused on the retrieval of documents, a unit of retrieval that frequently represents a partial solution to the information needs of the users. This assigns to the users the task of further analyzing documents from a provided ranking, in order to seek the exact answers to their questions. Furthermore, not only are verbose queries increasingly frequent (cf. Gupta and Bendersky [71][§1.2]), but also are entities more frequently mentioned in queries (cf. Bautin and Skiena [21]). Appropriately, entity-oriented search has been gaining relevance as an encompassing area of research [12], with multiple work unknowingly contributing to this larger area, either by focusing on semantic search[1], question answering, hybrid search, object retrieval, entity search, retrieval or ranking, or other generic approaches that leverage entities, such as document retrieval, sentence retrieval, or learning to rank.

4.1 Classical models

Some of the first approaches to entity-oriented search revolved around classical retrieval models, through the reuse of well-established text-based ranking techniques, as presented above. They include, most notably, defining virtual documents to represent entity profiles, or integrating results obtained from an inverted index and a triplestore.

Bautin and Skiena [21] presented what they considered to be the "first-in-literature" implementation of an entity search engine. Their first step was to find evidence that the task was relevant, based on the analysis of the AOL dataset, with 36 million web search queries. They found that 18–39% queries directly referenced entities and 73–87% contained at least one entity. They then proposed a concordance-based model for entity representation, along with an adaptation of Apache Lucene's[2] TF-IDF scoring scheme. Each concordance[3] (a virtual document) was built from the

[1] Semantic search as a task either refers to the semantically informed retrieval of documents, or to the retrieval of entities or relations over RDF graphs. We cover work on either approach, as both tasks are entity-oriented, using semantic search indiscriminately in both cases.

[2] `http://lucene.apache.org`

[3] A concordance is a list of terms and their context. In this case, the concordance is about entities and their context.

concatenation of all sentences containing the entity it represented, optionally for a given period of time (e.g., a month). Appropriately, they also proposed a time-dependent scoring function, modeling user interest in an entity as a function of time, and optimizing parameters based on the frequency of entities in the AOL query log. Finally, experiments were run over the entities extracted from an 18 GB collection of US news articles, collected through the Lydia pipeline [93]. They proposed a method for evaluating entity search by comparing the results list with the corresponding list obtained through a juxtaposition score [93]. The juxtaposition score measures the upper bound of the probability of two entities occurring in the same sentence under the assumption of independence. By obtaining the results list from Lucene and the results list based on the top related entities according to juxtaposition, the lists were then compared using the K_{min} distance from Fagin et al. [52], showing the best results for phrase queries with the slop parameter (word-based edit distance) equal to the number of query terms.

Bhagdev et al. [26] presented an example of hybrid search, where they combined keyword-based search with semantic search, showing that their approach outperformed either of the alternatives when individually used. They indexed text documents using Apache Solr[1]; they stored annotations generated by an information extraction system on a Sesame triplestore[2]; and they linked the extracted relations by annotating the provenance of the triples with the document of origin. At retrieval time, this enabled them to do keyword search over the inverted index, metadata search over the triplestore using SPARQL, and keywords-in-context search by retrieving text documents and matching them with triples through the provenance annotation. Their evaluation was based on 21 queries over a collection of 18 thousand technical documents. When comparing keyword search with metadata search, they obtained the best recall for keyword search (0.57 versus 0.40) and the best precision for metadata search (0.85 versus 0.56). However, when combining both approaches in a hybrid search, they obtained the best overall result, with a precision of 0.85 and a recall of 0.83. While the authors did not specifically mention it, this is clearly an example of entity-oriented search over combined data.

Pound et al. [118] proposed a formal model for ad hoc entity retrieval, but they used the designation *object* instead of *entity*, in the context of the web of data (the semantic web). They defined the task based on a keyword query for input, with an identifiable query type and query intent. The query was then processed over a data graph, returning a ranked list of resource identifiers (entities). Based on the analysis of real query logs from a commercial search engine, they also proposed five query categories for ad hoc entity retrieval: entity query, type query, attribute query, relation query, and other keyword query. These query categories can be mapped into specific tasks of entity-oriented search [12]. For instance, an entity or type query could be solved through ad hoc entity retrieval over virtual documents [21,120], while an attribute or relation query might be solved through related entity finding or entity list completion, if attributes were indexed as entities.

Koumenides and Shadbolt [88] proposed a Bayesian inference model for entity search. They combined link-based and content-based information as defined through RDF object properties and data properties. A query network was defined based on entity and property evidence, that could either be provided explicitly as entities or

[1] `http://lucene.apache.org/solr/`

[2] Sesame is now known as Eclipse RDF4J: `http://rdf4j.org/`.

implicitly as a combination of keywords. Common object or data properties were modeled through common identifier nodes O_i and D_j. By keeping separate nodes $o_{k,i}$ and $d_{k,j}$ for different instances of object and data properties, the model was able to use query nodes as evidence of object property identifiers, as well as data property identifiers or instances. This could then be further expanded into entities, or terms in the literal space. Unfortunately, the authors did not provide appropriate evaluation of their approach, making it unclear how it performs in relation to other approaches.

Urbain [139] presented a pipeline for entity-oriented sentence retrieval, proposing a strategy for the integration of terms (context), entities and their relations. He used a Markov network for modeling the dependencies between a pair of entities, a relation and a context, using a fully connected approach. No external knowledge bases were used. Instead, sentences in the form of triples $\langle$*entity*, *:relation*, *entity*$\rangle$ were obtained through natural language processing, extracting structure from documents and natural language queries. This enabled the construction of a Markov network that, together with user relevance feedback, was able to rank sentences by leveraging entities and relations. He compared several models, based on different combinations of feature functions for the Markov network. This included dependencies between entities, relations, and sentence and document terms. They consistently obtained better results for the proposed entity-relation model, supporting the importance of the entity graph in retrieval tasks.

Raviv et al. [120] proposed a general model for entity ranking, based on a Markov network for modeling the dependencies between the query and the entity. In particular, the model captured the dependencies between: (*i*) the entity document (i.e., a virtual document) and the query; (*ii*) the entity type and the query target type; (*iii*) the entity name and the query. A profile based approach, supported on a Dirichlet smoothed language model, was used for scoring entity documents. A filtering approach, based on the Kullback-Leibler divergence between the probability distributions of the entity and query types, was used for scoring the entity type. The entity name was scored using a voting or a global approach. The voting approach was based on the language models of retrieved entity documents relevant to the query. The global approach was based on the pointwise mutual information between the entity name and a query term. Evaluation was done over the INEX 2006 and 2009 Wikipedia collections, based on the topics and relevance judgments from the Ad Hoc track. In 2007, they obtained the best results, according to MAP, using full dependence over a ranking function based on the combination of the three dependency models. In 2008 and 2009, they obtained the best results, according to infMAP [147][§2.5], using sequential dependence for the same ranking function.

Raviv et al. [121] also tested the cluster hypothesis for entity-oriented search, i.e., the hypothesis that "closely associated entities tend to be relevant to the same requests". They experimented with four similarity metrics: (*i*) an exponential function of the shortest distance between any two categories of a pair of entities in the Wikipedia's category graph (*Tree*); (*ii*) the cosine similarity between the binary category vectors of the two entities (*SharedCat*); (*iii*) an exponential function of the negative cross entropy between the Dirichlet-smoothed unigram language model for the documents resulting from the concatenation of all the Wikipedia articles for each category (*CE*); and (*iv*) the cosine similarity between two vectors obtained from explicit semantic analysis (*ESA*). For each similarity measure, three different weighting schemes were used: L_{Doc}, $L_{Doc;Type}$ and $L_{Doc;Type;Name}$. For L_{Doc}, the Wikipedia

document corresponding to each entity was indexed and directly used to retrieve the entity. For $L_{Doc;Type}$, the similarity between the category set of each entity and the query target type was also taken into consideration. Finally, for $L_{Doc;Type;Name}$, the proximity between the query terms and the entity name was also taken into consideration. Evaluation was carried over the datasets for the 2007, 2008 and 2009 INEX Entity Ranking tracks, which used the English Wikipedia from 2006 and 2008. The authors found that the nearest neighbor cluster hypothesis holds. While result lists frequently contained 10-25% relevant entities, nearest neighbor entities of a relevant entity contained 30-53% relevant entities. Best results were achieved when using the *Tree* and *SharedCat* inter-entity similarity measures and were particularly good for the *Oracle* method, which employed cluster-based reranking based on the true percentage of relevant entities contained in each cluster. Other approaches included the *MeanScore* and *RegMeanScore*, which instead used the average score within a cluster of entities, optionally with regularization.

Bron et al. [31] tackled the task of entity list completion, where, given a textual description for a relation and a given set of example entities, the goal was to retrieve similar entities that respected the specified relation. Supported on language models, they experimented with text-based and structure-based approaches, as well as a combination of both. The text-based approach took advantage of the textual description of the relation, while the structure-based approach used the set of example entities provided as relevance feedback. For integrating both approaches, they experimented with a linear combination, as well as a switch method. The switch method was based on a performance overlap threshold, used to determine whether there was a relevant difference in performance between the two methods. In that case, they selected the method that achieved the highest average precision. Otherwise, when no relevant difference in performance was found, they simply relied on the linear combination. Their experiments showed that both approaches were effective, despite returning different results. They also found that the combination of the two approaches outperformed either one them when independently used. This further supports the need for a hybrid approach that combines both the strengths of text-based and structure-based features.

Bast and Buchhold [18] presented a novel index data structure for efficient semantic full-text search. They argued that neither classic inverted indexes nor triplestores could handle the problem individually. None of the approaches was able to provide multiple integration steps for different stages of query processing. They exemplified with a friendship relation that could only be found in the text, but should influence retrieved triples, potentially by establishing new connections. This was, however, unsupported by current approaches. Accordingly, they proposed a joint index for ontologies and text. As opposed to traditional keyword queries, they used trees as queries, based on the graphical interface provided by the Broccoli semantic search engine [17]. In order to provide a search mechanism over a tree query, the index distinguished between two types of lists: lists containing text postings, which they called context lists, and lists containing data from ontology relations. They evaluated efficiency, by comparing the inverted index and the triplestore baselines with two approaches (*Map*, linking context ID to entity postings, and *CL*, context lists with word and entity postings) based on their joint index. While the joint index supported all defined queries, these were only partially supported by each baseline individually, but completely supported by both when collectively considered. Overall, they found

the joint index approaches to require less disk space, taking similar or less time to query than the baselines.

Zhou [152] wrote a doctoral thesis on entity-oriented search, exploring the topic by distinguishing between querying by entities and querying for entities. In querying by entities, entities were taken as input, while results could either be documents or entities. In querying for entities, entities were returned as output, while queries could either be keywords or entities. He also highlighted the particular case of querying by and for entities, where entities were both taken as input and output. For querying by entities, he presented contributions on entity-centric document filtering. He proposed using an entity page, such as the associated Wikipedia page, to describe an entity in the query. This is different from the virtual document approach, described in previously covered work [21,120], in the sense that it is the entities in the query that are represented as documents, as opposed to the entities in the index. Regarding querying for entities, they proposed a content query language (CQL) over a relational-model based framework, as a solution to a data-oriented content query system. As opposed to keyword or entity queries, this querying approach required advanced technical knowledge, similar to SQL or SPARQL. In order to support CQL, they used an advanced index layer that included a joint index and a contextual index. The joint index combined pairs of keywords, keyword and data type, and pairs of data types, storing, for each occurrence, the document identifier, the position of the first keyword or data type and the distance to the second keyword or data type — only keywords or data types within a distance were considered for indexing.

Dietz and Schuhmacher [46] introduced Queripidia, as a set of knowledge portfolios. A knowledge portfolio represented a query-specific collection of relevant entities, combined with text passages from the web that explain why the entity is relevant to the query. They used two main datasets in order to develop a working prototype: the FACC1 entity link collection[1], a Freebase annotation of the ClueWeb corpora, automatically generated by Google; and the ClueWeb12[2], Category A dataset, used in the TREC Web track, where several test queries were also provided. Besides text passages, neighboring entities from the knowledge base were also included in the explanation, in order to provide additional context. In turn, each neighboring entity was associated with its own explanation in the context of the same query. This work is further detailed in Dietz et al. [47], where they explored several entity ranking approaches in order to understand whether the combination of documents and a knowledge base would improve entity ranking. All approaches were based on language models. They explored two different entity profile approaches: (*i*) using textual evidence surrounding the entity to establish context, and (*ii*) using the entity's Wikipedia page to represent the entity. The best retrieval performance was obtained based on the entity context, particularly for a window size of 50 words, when compared to the Wikipedia based approach. However, the best overall performance was achieved using a rank fusion technique based on the two methods, showing that the combination of text and knowledge in fact outperforms each individual approach.

[1] `http://lemurproject.org/clueweb09/FACC1/`

[2] `http://lemurproject.org/clueweb12/`

4.2 Learning-to-rank models

Chen et al. [38] explored the task of answer sentence retrieval, where sentences were ranked in respect to an input question. The challenge was that the best results did not necessarily contain the terms of the query, resulting in a lexical mismatch between the sentences and the question. This was an indicator that semantic features could be useful in tackling the problem. The authors proposed a learning to rank approach, establishing a baseline supported on Metzler-Kanungo (MK) features [101] — sentence length, sentence location, exact match of query in sentence, term overlap of query terms in sentence, synonym overlap of query terms in sentence, and language model (i.e., likelihood of query terms being generated by the sentence language model). They then proposed and tested two new semantic features, one based on ESA (explicit semantic analysis) [61] (the cosine similarity between the query and sentence ESA vectors), and another one based on the word2vec skip-gram approach [102] (the average cosine similarity between any query-word vector and any sentence-word vector). Through the evaluation of three learning-to-rank approaches — linear regression, coordinate ascent, and MART — they showed that results could be improved by leveraging semantic features. For each approach, they compared four feature configurations: (*i*) MK; (*ii*) MK + ESA, (*iii*) MK + word2vec and (*iv*) all features. The best results were consistently obtained for all features combined, except for MART, where MK + ESA obtained the best results, despite being closely followed by all features combined.

Lin et al. [92] tackled the task of related entity finding in TREC 2011 Entity track [14], where the goal was to rank the homepages of target entities, given a source entity, a target entity type and a narrative describing the relation between the source and target entities. Their approach consisted of document retrieval (using Yahoo!), entity extraction (using StanfordNER), feature extraction and entity ranking. For document retrieval, the goal was to obtain the homepage of an entity — their best approach was based on querying using the narrative to describe the relation. For entity ranking, they used a learning to rank approach based on features that considered frequency, density, proximity, semantic similarity, and the average rank of web pages, in regard to a candidate entity (e.g., total frequency of the entity in search results, similarity between the query and the entity type). They trained three SVM, one with default hyperparameters, another one with tuned hyperparameters, and a final one after applying feature selection. They discovered that the SVM with tuned hyperparameters performed better than the one with the default hyperparameters, and that the SVM with the selected features performed worse than the tuned SVM. Interestingly, they also discovered that directly using one of their proximity-based features yielded better results by itself. Based on the number of retrieved documents multiplied by the cumulative distance between the query and the entities in the documents, the authors were able to achieve better results than the SVM models. They also compared the tuned SVM with an approach based on a linear combination of all features, obtaining better results for the linear combination, thus finding that their assumption that the SVM would perform better was wrong.

Schuhmacher et al. [130] used a learning-to-rank approach for entity ranking, combining features about documents, entity mentions and knowledge base entities. They experimented with pairwise loss based on a support-vector machine, minimizing the number of discordant pairs in Kendall rank correlation coefficient. They also experimented with listwise loss based on coordinate ascent, minimizing both MAP

and NDCG. Several features were considered, based on an initial set of retrieved documents. In particular, they covered features like mention frequency, query-mention similarities, query-entity direct matching and path similarity over DBpedia, query term presence in the entity's Wikipedia article (based on a boolean retrieval model), the retrieval score for Wikipedia pages representing an entity (based on a sequential dependence model with Dirichlet smoothing), the PageRank of the entity's Wikipedia page, and entity-entity features measuring the path similarity between all considered entities (introduced in the model via a semantic smoothing kernel). Evaluation was carried over the REWQ datasets[1], created by the authors over the TREC Robust 2004 dataset and the ClueWeb12 corpus. They compared three baseline and three learning to rank models. The baseline models included the sequential dependence model, the mention frequency, and the PageRank. The learning to rank models included coordinate ascent and two SVMs, with and without a semantic kernel based on the relations between entities. They obtained the best overall results for the coordinate ascent approach. For the REWQ Robust dataset, the best performing individual feature was the sequential dependence model, while, for the REWQ ClueWeb12 dataset, it was the mention frequency. Both resulted in NDCG scores close to the learning to rank models.

Chen et al. [37] studied the effectiveness of learning to rank for ad hoc entity retrieval. They represented an entity based on a document with five fields derived from RDF triples: names, attributes (excluding the name), categories, related entity names and similar entity names (aliases). They then extracted query-entity features based on a language model, BM25, coordinate match, cosine similarity, a sequential dependence model (SDM) and a fielded sequential dependence model (FSDM). This resulted in a total of 26 features (five dimensions per feature, except for FSDM, which resulted in only one dimension). They experimented with a pairwise method (RankSVM) and a listwise method (coordinate ascent, optimized for MAP), comparing with the FSDM baseline, as well as a sequential dependence model and a mixture of language models, both optimized using coordinate ascent (SDM-CA and MLM-CA). They consistently obtained the best results for the two learning-to-rank approaches over test collections from well-known evaluation forums (SemSearch ES, ListSearch, INEX-LD and QALD-2). They also measured the influence of the fields and feature groups in the RankSVM approach, overall finding that the related entity names was frequently an important field, and that the SDM related features were in general the most influential.

Gysel et al. [73] have tackled the problem of product search based on representation learning. They proposed the latent semantic entities (LSE) for jointly learning the representations of words (W_v), entities (W_e), and a mapping between the two (W). A string, be it an n-gram from a document or a keyword query, is mapped to the entity space based on the following steps. Given a word represented by its one-hot vector, a learned matrix W_v of word embeddings is used to map the averaged one-hot vectors of the string to its embedding. A word embedding is then mapped to the entity space using a learned matrix W and bias vector b and applying the *tanh* function. An entity can also be represented in the same space, based on its embedding, as defined in the entity embeddings matrix W_e. Learning is done based on gradient descent over a loss function $L(W_v, W_e, W, b)$. They evaluated the effectiveness of LSE in an entity retrieval set based on a learning-to-rank pairwise

[1] `http://mschuhma.github.io/rewq/`

approach (RankSVM), exploring query-independent features (QI), a query-likelihood language model (QLM), and the latent semantic entity representation (LSE). Their best results were consistently obtained for QI + QLM + LSE, tested over different product categories, when compared to QI, QI + QLM, and QI + LSE.

5 Graph-based models

Search is based on a simple principle developed in the library. In order to find a relevant page of a book, based on a given keyword, we originally had to scan the book, page by page. This was a time consuming task, particularly for books with a large number of pages. The problem was solved through the back-of-the-book index, where a list of manually selected keywords would point to the pages mentioning a given concept. Taking only a few pages and using an alphabetical order, this approach was more efficient than reading the whole book. The same principle applies when indexing a collection of documents in a computer. A collection that would take a long time to be fully scanned is condensed in an inverted index, where terms point to lists of documents, storing statistics like the frequency or the positions of the term in the document. As opposed to the back-of-the-book index, an inverted index contains most of the terms in the collection, usually discarding frequent words (stopwords) and sometimes storing a reduced form of the word (obtained from stemming or lemmatization). Automatization means that a larger volume of data can be processed efficiently, and stored statistics can be used as a way to measure relevance. However, one thing that is lost with the inverted index is the ability to relate concepts. In the back-of-the-book index, a domain expert might provide associations between concepts (e.g., using 'see also') or use keywords that are not explicitly mentioned in the page despite being more adequate for search. The inverted index is usually focused on representing the document as is, however we can use techniques like query expansion or latent semantic indexing to establish new connections that make documents more findable. With query expansion we can, for instance, also consider the synonyms of the query keywords to increase recall. With latent semantic indexing we can establish new relations based on contextual similarity, or we can use approaches like word2vec or explicit semantic analysis for a similar purpose.

Another relevant source of concept relations are knowledge bases, which are more explicit and can be used to improve retrieval by leveraging the semantics provided by entities. Due to the complex relations between entities, knowledge bases are usually represented as graphs. The most frequently used model for this is RDF (resource description framework), a tripartite labeled directed multigraph. In an RDF graph, each relation is modeled by three linked nodes known as a triple — a subject (entity), a property (relation), and an object (entity or attribute). Other approaches include topics maps or the property graph model. Topic maps model topics through their associations and occurrences. Topics are analogous to keywords in the back-of-the-book index, while occurrences are analogous to the page numbers. Associations can represent n-ary connections between topics, similar to the role of the 'see also' expression in the back-of-the-book index. In the property graph model, relations are captured between entities, but properties are not explicitly a part of the graph, being externally associated with nodes and edges instead. In comparison to RDF, attributes and relations are not represented as nodes in the graph, but are instead stored in a node property index and defined as edge labels, respectively. RDF is a strong model for

inference, while the property graph model provides a solid base for ranking entities without having to consider the effect of tripartite relations or having to compute a projection over one the three modes. Knowledge graphs [12][§1.4.4] are usually queried through a structured language like SPARQL, used for graph matching. Unlike unstructured keyword-based queries, SPARQL is not user-friendly, in the sense that it requires a certain degree of technical expertise that is more distant from natural language. There is a need for keyword-based retrieval over knowledge graphs, but also for the structured data that knowledge graphs usefully provide to improve the effectiveness of document retrieval. Furthermore, understanding graph-based models for representing, retrieving or otherwise manipulating text and/or knowledge is an essential step towards providing a solution for general information retrieval. On one side, graphs are ideal for dealing with the problem of heterogeneity [54]. On the other side, and perhaps more importantly, awareness about a diverse set of graph-based models, from multiple application contexts, is essential to support the quest for finding a joint representation model of terms, entities and their relations, along with a universal ranking function that can be used for entity-oriented search and, eventually, for information retrieval in general.

Many of the graph-based techniques currently applied to entity-oriented search, were surveyed in 2005 by Getoor and Diehl [66], who grouped them into the area of link mining[1]. They covered tasks from link analysis, community detection, entity linking, and link prediction that, in some way, provide a workbench for developing graph-based entity-oriented search. In this section, we survey the usage of graph-based models for multiple retrieval tasks, from modeling documents as graphs, to providing query-dependent and query-independent evidence of document or entity relevance. In Section 5.1, we present classical link analysis approaches, covering PageRank, HITS and heat kernel. In Section 5.2, we introduce graph-based representations of documents, used for ad hoc document retrieval. In Section 5.3, we present retrieval methods based on knowledge graphs, for improving or augmenting document retrieval, as well as for entity retrieval. In Section 5.4, we explore approaches that rely on entity graphs built directly from text corpora, and in Section 5.5 we cover tensor based approaches for representing entity graphs. In Section 5.7, we provide an overview on hypergraph-based models, covering tangential work with applications to entity-oriented search. Finally, in Section 5.8, we focus on random walk based models, in particular covering applications of PageRank to entity-oriented related tasks.

5.1 Link analysis

Classical graph-based models in information retrieval include HITS and PageRank, two link analysis algorithms developed to rank pages in the web graph. In 1999, Kleinberg [86] proposed the hypertext induced topic selection algorithm (HITS) as a combination of an authority score, based on incoming links, and a hub score, based on outgoing links. The computation of HITS is frequently done over a query-dependent graph, built from a root set of pages that are relevant to the query. The root set can be retrieved using a classical model like TF-IDF or BM25 and it is then expanded

[1] There is not much evidence of link mining as an area beyond this survey, which leads us to believe that, albeit a good one, this showed no relevant adoption by the community.

into a base set that includes all outgoing links and a subset of incoming links. While the number of outgoing links is usually small, the number of incoming links can be too high for an efficient computation. Thus, a parameter d is used to define a ceiling for the number of incoming links to consider. When the number of incoming links surpasses d, then only a random sample of size d is considered, otherwise all incoming links are considered. In its query-dependent application, HITS is more expensive than PageRank for ranking, since it cannot be computed offline. Like PageRank, HITS is also related to the leading eigenvector of a matrix derived from the adjacency matrix. Interestingly, the authority and hub scores are related to the leading eigenvectors of AA^T and A^TA, respectively, both sharing the same eigenvalue [127][§3.2].

Also in 1999, Page and Brin [116] proposed PageRank as a way to measure the importance of web pages. PageRank [30] is an elegant algorithm that offers multiple interpretations and computation approaches. It can be seen as the solution to a linear system [67, 42], or as the eigenvector of the Markov chain derived from the graph — after adding a teleportation term to the transition probabilities, in order to deal with sinks (i.e., pages without any links to other pages). It can be solved through Gaussian elimination, power iteration or even Monte Carlo methods [9]. Conceptually, PageRank is a random surfer model, where the probability of visiting a node reflects the behavior of a user that is randomly navigating the web by clicking hyperlinks, while occasionally jumping to a new page. This model is recursive, in the sense that it results in a centrality metric where the importance of a node depends on the importance of its neighbors — the better connected a node is, both through quantity (i.e., many nodes) and quality (i.e., nodes that are themselves well connected), the higher the PageRank. Research about PageRank has led to many applications [68], exploring contextual information (e.g., Topic-Sensitive PageRank [76]), combinations of features (e.g., Weighted PageRank [48]), alternative smoothing approaches (e.g., Dirichlet PageRank [141]) or historical evidence (e.g., Multilinear PageRank [69]). One of the variants, Reverse PageRank [57], consists of simply reversing the edge direction and computing PageRank for this complementary graph. It is to PageRank what the hub score is to the authority score in HITS. Bar-Yossef and Mashiach [15] have shown that the Reverse PageRank is not only useful to select good seeds for TrustRank [72] and for web crawling, but also, more interestingly, for capturing the semantic relatedness between concepts in a taxonomy. According to Gleich [68][§3.2], Reverse PageRank can be used to determine *why* a node is important, as opposed to simply identifying *which* nodes are important, something that PageRank already solves. The success of PageRank in complementing itself through different applications is a sign of the usefulness of random walks in solving diverse tasks, which is a useful characteristic in the design of general models.

Node importance is generally measured based on the number of incoming links (as we have seen with HITS authority and PageRank) or based on the favorable structural position of a node (e.g., closeness [22], betweenness [60]). Besides node importance, node relatedness can also be measured as a type of structural similarity, usually based on whether two nodes share links to or from a common node. Van and Beigbeder [140] explored the effect of node relatedness in the retrieval of scientific papers based on a user profile. They experimented with bibliographic coupling and co-citation as reranking strategies. In bibliographic coupling, two papers are related if they cite a common publication. In co-citation, two papers are related if they are cited by a common publication. For measuring co-citation, they implicitly built a graph based on Google search results for pairs of paper titles, as well as based on

data from the Web of Science. Based on the 20 content-only topics from INEX 2005, each representing an information need of a user, the authors selected approximately five papers per topic to establish a user profile. Using Zettair[1], they then indexed the collection of papers, ignoring those used to build user profiles. They retrieved 300 papers for the 20 topics, based on Dirichlet-smoothed language models, and used this as the baseline. Results were then reranked based on bibliographic coupling, co-citation using the Web of Science, and co-citation using Google. They obtained a consistent improvement over the baseline only for the Google-based co-citation reranking (P@10 increased from 0.62 to 0.68).

Link analysis can also be approached through kernels, supporting both the measurement of importance and relatedness. Ito et al. [82] explored von Neumann kernels as a unified framework for measuring importance and relatedness, using different parameter configurations to go from co-citation or bibliographic coupling ($n = 1$) to HITS (large values of n). They also identified two limitations of co-citation relatedness: (*i*) two nodes are considered to be related only when they are cited by a common node; (*ii*) relatedness only takes into account the number of nodes commonly citing two nodes, as opposed to also considering the differences in popularity of the two nodes (e.g., co-citing a generic web site and Google might not be an indicator of relatedness, given the popularity of Google). As a solution, they proposed the use of Laplacian and heat kernels, which enabled them to control the bias between relatedness and importance, while effectively mitigating the identified limitations.

5.2 Text as a graph

For unstructured text, without hyperlinks, there are also models to represent documents as a graph of words. Blanco and Lioma [27] provide an in-depth exploration of graph-based models for text-based retrieval. They defined two graph-based representations of terms in a document, based on an undirected and a directed graph. The undirected graph linked co-occurring terms within a window of size N. Similarly, the directed graph also linked co-occurring terms within a window of size N, but established a direction based on grammatical constraints. This required POS tagging to be applied to terms and then, based on Jespersen's rank theory [83], POS tags were assigned a degree — 1st degree for nouns, 2nd degree for verbs and adjectives, 3rd degree for adverbs (and 4th degree for other tags). Under this model, higher rank words can only modify lower rank words. This relation was captured using a directed edge in the graph. Two raw metrics were then defined over each graph, using PageRank and the (in)degree to weight term nodes. This resulted in TextRank and TextLink over the co-occurrence graph (undirected), and PosRank and PosLink over the co-occurrence graph with grammatical constraints (directed). They then combined each raw term weighting metric with IDF for ranking documents according to the terms of a given query. This raw model was combined with several individual graph-based features, using the *satu* method by Craswell et al. [43], and retrieval effectiveness was assessed over TREC test collections (DISK4&5, WT2G and BLOGS06). Graph-based features added to the raw model included: average degree, average path length, clustering coefficient, and the sum of graph-based term weights (which worked as a type of document length normalization). The graph-based models were compared to

[1] `http://www.seg.rmit.edu.au/zettair/`

the BM25 (the baseline), as well as TF-IDF, according to MAP, P@10 and BPREF (binary preference). The best results for graph-based models were obtained for the BLOGS06 collection. Generically, the graph-based features improved the raw model and there was always a graph-based model that outperformed the baseline, although for some of them the difference was not statistically significant. They also measured the impact of the window size N, finding that $N = 10$ performed well for MAP and BPREF, and they measured the impact on indexing time introduced by computing the graph-based features, finding that TextRank only introduced an overhead of a few milliseconds ($\sim 50ms$ for 1,000 iterations).

Building on the previous work, Rousseau and Vazirgiannis [126] proposed a novel graph-based document representation, defying the term independence assumption of the bag-of-word approach. They defined an unweighted directed graph (the graph-of-word), where nodes represented terms, and edges linked each term to its following terms within a sliding window of size N, in order to capture context. Based on information retrieval heuristics [53,97] and the graph-based term weighting approach by Blanco and Lioma [27], they also defined a retrieval model over the graph-of-word, based on the indegree of the nodes (TW-IDF). The goal of the weighting model was to measure the number of contexts a given term appeared in. They also introduced a pivoted document length normalization component, tunable with parameter b (analogous to BM25's b). The graph-of-word was generated per document, computing the TW metric and storing it within the inverted index, to be used as a replacement for TF. This meant that the document graphs could then be discarded without requiring persistence. They evaluated the TW-IDF ranking function with and without regularization over document length, as well as with and without parameter tuning for the pivoted document length normalization b parameter. They found that only a small contribution of document length normalization was required, thus settling on a constant value of $b = 0.003$. They also experimented with parameterizing the window size N, but since they didn't find an improvement for any of the tested values, they used a default value of $N = 4$. Finally, they did a comparison of TW-IDF with TF-IDF and BM25, as well as Piv+ and BM25+ (TF-IDF and BM25 with lower bound regularization [97]), showing that TW-IDF consistently outperformed the other weighting functions, particularly in realistic conditions, where parameter tuning is costly and is seldom an option.

Dourado et al. [49] have come forth with a general graph-based model for text representation, able to support both the tasks of classification and retrieval. Their approach consisted of mapping text documents to a directed graph of words, capturing term order, and assigning node weights based on the normalized TF of the terms and edge weights based on normalized TF of bigrams formed by the two words represented by the linked nodes. This is done for the whole collection, document by document. For each document, subgraphs are then extracted, for example based on segments within a given path length, and then a vocabulary selection stage is carried based on a graph dissimilarity function and on graph clustering. Each cluster corresponds to a word (a centroid) in a codebook, representing the vocabulary that will be used to represent the documents. The subgraphs in each document graph will then be assigned to a centroid, either by hard assignment (the closest centroid), or soft assignment (based on a kernel function). The output of the assignment function is a matrix, where each vector represents the assignment weight to each centroid. A final pooling function then collapses this matrix into a vector that represents the document graph, reaching the goal of graph embedding. From this point on, the vec-

tor can be used both for retrieval or classification, which the authors evaluated using multiple test collections. For the task of classification, their bag of textual graphs approach (as they called it), was able to outperform the remaining document representations for four of the five test collections, according to macro F1, which ranged from 0.676 to 0.997. For the task of retrieval, they experimented with the bag of textual graphs using three distances: Euclidean, Jaccard index, and cosine. They were able to outperform all baseline approaches, according to NDCG@10, when using the Jaccard and cosine distances, and most of them when using the Euclidean distance. The best results were obtained for the cosine distance.

Recent work by Gerritse et al. [65] has focused on exploring graph-embedding for improving entity-oriented search tasks. They did this in two stages, first by using state-of-the-art retrieval models (BM25F and FSDM), and then by reranking based on the embedding space. In particular, they compared the usage of graph-embeddings and word embeddings based on Wikipedia data, showing that relying on the link graph was fundamental for computing the embeddings and approximating the the cluster hypothesis. This ensured that similar entities were grouped close together but far apart from groups of dissimilar entities, leading to well-defined clusters and improved retrieval effectiveness. They carried an experiment based on the DBpedia-Entity v2 collection, using NDCG@10 and NDCG@100 for evaluation. Both for the reranking over FSDM and BM25F, there was a clear and consistent improvement for the version that considered the Wikipedia link graph in the computation of the embeddings.

In another recent contribution, Irrera and Silvello [81] have proposed a complete pipeline, from entity linking to ranking, where they used graph-based features and a learning-to-rank model, in order to solve the background linking task from the TREC News track. A graph was created per document, based on the semantic relatedness between the entities extracted from the text. They applied pruning to the graph, keeping only the largest community of the largest connected component. Then, several document-based and query-based features were extracted from the text, as well as from the graph, which were used to train a model based on list-wise loss. Several hyperparameter configurations were tested, each resulting in a differently optimized model, and distinct computed rankings were fused to obtain new and improved scores. Evaluation was done by computing the reciprocal rank, P@1 and NDCG@1, showing improvements over a BM25 baseline. For different cutoff values of NDCG (@5, @10 and @100), only NDCG@5 was higher for the proposed model when compared to BM25, showing that performance particularly improved within the top 5 results, when using learning to rank with text-based and entity graph-based features.

5.3 Knowledge graphs

Instead of issuing direct queries over a graph, either by ranking its nodes (Section 5.1) or by matching subgraphs (Section 5.6), graph-based models can simply be used for the representation of knowledge in a retrieval process. We also consider such approaches to be graph-based, as long as there is an obvious and direct dependence on the entities and relations in a knowledge graph.

Knowledge graphs have multiple applications. Zou [155] provides a short and focused survey that covers the purpose of these semantic structures in areas like

question answering, recommender systems, or information retrieval, also covering domain-specific and other applications. The wide range of domains that can benefit from this graph-based model partly illustrates the ability for this data structure to be used in a general manner to unify information in a practical environment.

5.3.1 Augmenting entities with documents and vice-versa

Fernández et al. [55] showed that ontology-based semantic search can be used for augmenting and improving keyword-based search. They proposed a system architecture for question answering based on natural language queries over the semantic web, using ranked documents to complement an answer given by ranked triples. The system relied on an ontology index, a concept-based index and a document index. The ontology index mapped terms to entities and was used both to build the concept-based index (document annotation) and for query processing (query annotation and triple matching). In particular, the PowerAqua system [94] was used for mapping keywords in a natural language query into triples from the indexed ontologies — they relied on WordNet to improve the matching between query terms and entities. The document index mapped terms to documents and was used for document ranking based on the retrieved triples and the concept-based index. Evaluation was performed using the TREC WT10G collection and a selection of 20 topics and their relevance judgments from TREC9 and TREC 2001. They also relied on 40 ontologies, based on Wikipedia, that covered the domain of the selected topics. Each TREC topic was expanded with an appropriate question answering request and additional information on available ontologies. They experimented with a baseline using a text-based approach over Lucene, semantic query expansion based on PowerAqua, and their complete semantic retrieval approach. When compared to the baseline, they obtained an improved effectiveness for 65% of the evaluated queries, according to average precision and P@10, when using their semantic retrieval approach, and 75% when considering only P@10 and either of the semantic approaches.

Byrne [33] dedicated her thesis to exploring the unified representation of hybrid datasets, combining structured and unstructured data, particularly in the domain of digital archives for cultural heritage. She relied on RDF triples, with a subject, predicate and object, to generate a graph that would integrate structured data from relational databases, unstructured data from entities and relations extracted from free text, and even domain thesauri useful for query expansion. For relational databases, each row in a table was instanced as a blank node of a class with the table name. For domain thesauri, the SKOS ontology was used to represent concepts and their relations of synonymy or hyponymy. For free text, 11 entity classes were considered, along with 7 predicates, one of which had a higher arity, containing 6 subpredicates that were used to establish binary relations. A classifier was trained for named entity recognition, and another one for relation extraction. Finally, equivalent queries were prepared to run over the RDF store as SPARQL, running either within Jena or AllegroGraph, and over the relational database as SQL, running within Oracle or MySQL. Byrne found that queries over RDF were considerably less efficient than queries over relational databases. She also found a lack of aggregation functions like *count* or *average* to query RDF, as well as the lack of graph theory functions to identify node degree or shortest paths.

Balog et al. [13] presented the SaHaRa entity-oriented search system for searching over news collections. For a given keyword query, SaHaRa used language models to

retrieve both documents and entities, displaying them in a two-column interface. A document-centric view and an entity-centric view were also provided. The document-centric view was used to display a news article along with links to related articles and associated entities. The entity-centric view was used to display the entity, showing for example its Wikipedia summary, along with links to related news and Wikipedia articles, as well as associated entities, either based on the language model or the DBpedia relations. SaHaRa illustrates the benefits of augmenting documents with entities, as well as entities with documents, also showing that language models can be used for documents as well as entities.

5.3.2 Text-based retrieval of entities

Blanco et al. [28] tackled the problem of effectiveness and efficiency in ad hoc entity retrieval over RDF data. Their ranking approach was based on BM25F, experimenting with three representation models: (*i*) an horizontal index, where fields *token*, *property* and *subject* respectively stored terms, RDF property names, and terms from the subject URI; (*ii*) a vertical index, where each field represented a separate RDF property name (e.g., *foaf:name*) containing terms from the respective literals; and (*iii*) a reduced version of the vertical index where fields represented *important*, *neutral* and *unimportant* values depending on the classification of the corresponding RDF properties. Evaluation was carried over the Billion Triple Challenge 2009 dataset [77]. For measuring effectiveness, they used the 92 entity-oriented topics and relevance judgments from the Semantic Search Challenge of 2010, obtained from Microsoft Live Search query logs. They compared BM25 from MG4J[1] with the three proposed indexes, finding the horizontal index to be the least efficient for *AND* and *OR* operators. Both the vertical and the reduced-vertical indexes were able to obtain a lower but comparable performance to BM25 for the *AND* operator, but not for the *OR* operator. Efficiency-wise, the best RDF index was the reduced-vertical. Regarding effectiveness, they compared BM25F with the BM25 baseline, as well as the best performing submission for SemSearch 2010. They found that, while the BM25 baseline was worse than the SemSearch 2010 baseline, their BM25F approach was able to improve MAP in 42% and NDCG in 52%. BM25F's b, field weight and document weight parameters were optimized using linear search and the promising directions algorithm [125], increasing MAP in over 35% just for tuning the parameter b for each field. Increasing the weight of documents from *important* domains (e.g., dbpedia.org) was also significant.

Neumayer et al. [106] presented an overview on entity representations for the text-based retrieval of entities. They covered the unstructured entity model, where all textual evidence was aggregated as a field in a virtual document, as well as the structured entity model, where textual evidence was aggregated in multiple fields, one per predicate type, in a virtual document. In particular, the aggregation into four predicate types was suggested: *Name*, *Attributes*, *OutRelations* and *InRelations*. Language models could then be applied to either representation and used as a ranking function, either over a single field or over the four individual fields. The presented models did not, however, preserve or take advantage of the information provided by individual predicates. Accordingly, the authors proposed the hierarchical entity

[1] `http://mg4j.di.unimi.it/`

model, where an entity was represented by the predicate types, as well the corresponding predicates. Additionally, each predicate type was represented both by its predicates and the text evidence for the type, and each predicate was represented by the text evidence for the predicate.
They also proposed four approaches for predicate generation $P(p|p_t,e)$ — *Uniform* (inverse frequency of predicates of the given type), *Length* (number of terms per predicate, normalized for the length of its predicate type), *Average length* (average number of terms per predicate, normalized for the average number of terms of its predicate type) and *Popularity* (fraction of triples with a given predicate, normalized for the number of triples containing any predicate of the same type). They found that the hierarchical entity model was able to outperform the unstructured entity model, but not the structured entity model. Perhaps more interestingly is that fact that it was able to fully capture the original semantic relations, without incurring in a significant loss of performance.

Oza and Dietz [114] explored several types of entity relations and how they affected retrieval effectiveness, finding co-occurrence to be the best type of relation to be taken into account when constructing a story. They relied on a large-scale benchmark from the TREC Complex Answer Retrieval, computing MAP, Rprecision and F1 for evaluating 12 features, as well as a learning to rank model combining some of those features. The features included: (*i*) a relevance score for the co-occurrence of entities, (*ii*) the simple co-occurrence count, and (*iii*) the mention frequency in retrieved passages, as well as (*iv*) the number of outlinks, (*v*) inlinks, (*vi*) bidirectional links, and (*vii*) undirected links, between entities in the knowledge base (accumulated for each entity by incident links), (*viii*) bibliographic coupling and (*ix*) co-coupling (i.e., number of outlinks and inlinks shared by two entities), (*x*) relevance weighted bibliographic coupling and (*xi*) co-coupling (i.e., accumulating relevance scores instead of counting the outlinks and inlinks), and (*xii*) a simple BM25 score computed based on the entity description present in the knowledge base. In their experiments, feature (*i*) always achieved the best MAP, Rprecision and F1, only being surpassed by a learning-to-rank model over features (*i*), (*ii*), (*iii*), and (*vii*).

5.4 Entity graph from text

We have seen that both unstructured text and structured knowledge can be modeled as a graph. Beyond these individual representations, there are also approaches that focus on building entity graphs from text, establishing a direct relation between text and knowledge. This also helps to distinguish between knowledge that is internal and knowledge that is external to the collection. Bordino et al. [29] explored the topic of serendipity in entity search, evaluating results based on surprise and relevance, as well as based on interestingness. They created an entity network from Wikipedia and Yahoo! Answers based on the similarity of entities profiles built from the textual content citing an entity. In order to improve performance, they only compared pairs of entities that co-occurred in at least one document, based on the document similarity self-join algorithm by Baraglia et al. [16]. They then created an edge between two entities when their similarity was above a given threshold. For evaluation, they collected the most searched queries in 2010 and 2011 from Google Trends[1], identify-

[1] Google Trends is identified in the paper as Google Zeitgeist, which was a previous designation.

ing the entity associated with each query. The queries covered topics about people, places, websites, events, gadgets, sports, and health. They then used a crowdsourcing platform to obtain relevance judgments and indicators of interestingness, and they quantified surprise based on whether results appeared on commercial search engines, according to different criteria. Finally, serendipity was measured based on the normalized aggregated relevance of surprising results. They found that 51% of the nodes in the Wikipedia network overlapped with the nodes in the Yahoo! Answers network and also that both networks were nearly 95% connected, through the presence of common concepts that bridged the gaps. Their ranking method was based on the stationary (time-independent) distribution of lazy random walks in the graph, with a $\lambda = 0.9$ probability to stay in the input entity node, which at $d = 0.85$ worsened the results[1]. They also introduced three main constraints based on: (*i*) quality (measured through readability); (*ii*) sentiment (based on SentiStrength[2] as applied to the associated textual documents); and (*iii*) topic categories (using a proprietary classifier to identify 18 main categories). They then measured the fraction of unexpected (surprising) and relevant recommendations, over different runs, for unconstrained search, as well as considering topic, high sentiment, low sentiment, high readability, and low readability constraints. Overall, they obtained the best results for topic constrained search and for high readability constrained search. They showed that Wikipedia and Yahoo! Answers were good datasets for promoting serendipitous search, as they returned relevant results that were dissimilar to those found through other web search engines. Recommendation tasks as the one presented in this work are analogous to the entity-oriented search task of related entity finding, although ignoring the target entity type and experimenting with additional constraints.

Ni et al. [107] proposed a concept graph[3] representation of a document and the measurement of semantic similarity based on that graph. They used TAGME[4] to annotate documents with mentions linked to Wikipedia concepts. Then they built a graph using concepts as nodes and three types of concept relations as edges — *:context* (connecting concepts sharing incoming links from common Wikipedia articles), *:category* (connecting concepts belonging to similar categories in the Wikipedia taxonomy), and *:structure* (based on the graph induced by the links within the Wikipedia infobox of each concept and the shortest path between concepts). The *:context* and *:category* edges are similar to the bibliographic coupling and co-citation approaches described in Section 5.1, based on the work by Van and Beigbeder [140] to capture semantic relatedness. Each edge was weighted by a similarity metric proposed for each specific concept relation type. The authors also assigned weights to the nodes based on the closeness centrality for each node, using a custom distance metric defined by the inverse of a linear combination of the weights of the three possible types of edges. An additional weight was associated with each node, based on the TF-IDF similarity between the concept's Wikipedia article and the represented document. Then they defined a pairwise concept similarity, called *Concept2VecSim*, where *Concept2Vector* was inspired by word2vec, and a document similarity called *ConceptGraphSim* based on the best pairwise similarities of each concept of either

[1] Please notice that we normalized the notation to be consistent over the document. Here, $\lambda = \beta$ and $d = 1 - \alpha$, when compared to the original paper.

[2] `http://sentistrength.wlv.ac.uk/`.

[3] Not to be confused with conceptual graphs [135].

[4] `https://tagme.d4science.org/tagme/`.

document, relative to the concepts in the other document, as well as the weight of the concepts in the graph. They compared their methodology, optionally combined with ESA [61], with several other state-of-the-art methodologies, both through individual and combined applications. They concluded that their approach outperformed the majority of the methodologies, with the exception of *WikiWalk + ESA* [145] when compared with *ConceptGraphSim* alone, and *ConceptsLearned* [79] when compared with *ConceptGraphSim + ESA*.

5.5 Entity graph as a tensor

Zhiltsov and Agichtein [149] captured the latent semantics of entity-relations based on tensor factorization. They defined a tensor that described entity relations based on different predicates, represented as multiple adjacency matrices, one per predicate over the third dimension of the tensor. Tensor factorization was applied to the tensor, using the RESCAL algorithm [108], in order to obtain a matrix of latent entity embeddings and a tensor of latent factors. A listwise learning to rank approach, based on gradient boosted regression trees, was then used to optimize a ranking function according to NDCG. They considered term based features, as well as structural features. Term based features relied on a multi-field document representation of the entity, enabling the retrieval of entities based on keyword queries. In particular, they did this based on a mixture of language models, as well as a bigram relevance score per field. Structural features were based on the entity embeddings from the tensor. In particular, they computed the cosine similarity, the Euclidean distance and the heat kernel between the embedding of a given entity and the embeddings of each of the entities in the top-3 using a baseline ranking. Their evaluation was based on 142 queries from the SemSearch Challenge from 2010 and 2011 and the Billion Triple Challenge 2009 dataset. They consistently obtained an increased performance of nearly 5%, for NDCG, MAP and P@10, when considering structural features.

5.6 Graph matching

One of the approaches to graph-based retrieval is the definition of graph queries (e.g., translated from keywords or natural language), that can be issued over a text graph or a knowledge graph. In the context of graph data management, Fletcher et al. [56][§1.4.1] classified graph queries into four categories: adjacency queries, pattern matching queries, reachability queries, and analytical queries. Adjacency queries consider nodes linked by an edge, as well as edges that share a common node, and they can even consider a k-neighborhood (i.e., linked nodes/edges at a distance k). Pattern matching queries consist of finding values for variables in a triple or sequence of triples (e.g., $\langle ?x, :friend, ?y \rangle$ should return pairs of friends). Reachability queries determine which nodes can be reached based on the given traversal restrictions (e.g., $\langle John, :friend^{+}, ?x \rangle$ will return friends of *John*, as well as friends-of-friends of *John*, and so on). Finally, analytical queries include queries that are based on aggregated computations over a graph, including average path length, connected components, community detection, clustering coefficient, or PageRank. Fletcher et al. [56][Ch.4] also covered the concepts of query relaxation and approximation as a way to manipulate the path structure in a graph query to enable a more flexible query processing.

This is aligned with the need for better retrieval techniques over knowledge graphs that, unlike text-based retrieval, do not yet provide adequate approaches based on keyword or natural language queries. Entity-oriented search tackles this type of challenges, making search easier over unstructured and structured data.

Zhu et al. [153] and Zhong et al. [150] have proposed an approach to semantic search for entity ranking, through the matching of a query graph and a resource graph. The idea was developed based on conceptual graphs [135], having a direct translation to RDF graphs[1]. The conceptual graphs were built from natural language queries and documents via their prototype ALPHA [91]. They measured the similarity between two conceptual graphs based on the similarity between their nodes and edges. Node similarity was computed using WordNet[2], based on the distance to the closest common parent of two concepts. Concepts that subsumed each other were considered to have distance zero and thus similarity one. Edge similarity was computed as a binary value that was one, only when the edge from the query graph subsumed the edge from the resource graph. For the computation of graph similarity, they avoided the maximum subgraph matching problem, which is NP-complete, by defining entry nodes that the user should identify in their queries.

Minkov and Cohen [103] were concerned with personal information management and the application of graph walks to derive entity similarity. They were able to use queries to generalize multiple tasks over an entity-relation graph (e.g., modeling e-mail as a graph of *people* who send and receive *messages* that contain *terms*). A keyword query was first processed in order to identify corresponding nodes in the graph, along with the target type of the output nodes. A response to the query consisted of a ranked list of entities of the given target type. Different tasks were defined based on the relations in the graph and could be specified along with the query. User feedback was also considered for learning task-specific similarities. This approach fits the tasks of related entity finding or entity list completion in entity-oriented search. Graph walks were based on personalized PageRank and, in particular, the alternative notation already presented in the context of the heat kernels used by Chung [40] and Kloster and Gleich [87]. They considered introducing walk bias based on the learned edge weights on a per-task basis. They also considered a reranking approach based on global features of the graph, such as the reachability from seed nodes (i.e., the count of source nodes that link to target nodes). They evaluated several personal information tasks, based on MAP and P@1, modeling them as queries over the graph. Overall, the best results were obtained for the reranked version of the graph walks, with the exception of one dataset where the learned weights version performed better. For the threading task, they used the TF-IDF as the baseline, overall obtaining the best results for the reranked version of the graph walks. Finally, for the alias finding task, they also relied on the Jaro similarity score for the baseline, obtaining the best results for the graph walk.

Zhong et al. [151] have worked on keyword-based search over knowledge graphs. They combined content-based and structure-based features to score answer trees. In particular, weights were manually assigned to nodes, depending on the context, and then PageRank was used to balance and normalize the weights. Initial weights were also assigned to edges, computing their final weight based on the weights of the source and target nodes, as well as the initial weight. Answer trees were then

[1] `https://www.w3.org/DesignIssues/CG.html`

[2] `https://wordnet.princeton.edu/`

extracted from the graph based on whether they contained the query keywords, and scored based on the distances between the root node and each query keyword. The distances were multiplied by a penalizing factor, that increased with the number of previous trees with the same root node, and then they were summed. Evaluation was done over a DBLP[1] graph with 840 thousand vertices, 1.3 million edges, 95 thousand terms, and edge weights between 0.31 and 0.99. Their approach to evaluation was based on a manually built collection of keyword queries and a manual assessment of the rankings, with a focus on the diversity of the returned results.

Zhu et al. [154] proposed a natural language interface to a graph-based bibliographic information retrieval system. Through named entity recognition and dependency parsing, they were able to generate a graph query that was capable of correctly interpreting 39 out of 40 natural language queries of varied complexities. The approach relied on a graph database to store the bibliographic data. A natural language query was then processed using named entity recognition to obtain the nodes for the graph query to be issued over the graph database. Dependency parsing was applied to extract relations between tokens (including entities), which were then adapted to the database schema, for instance adding missing nodes (e.g., the dependency ⟨*papers*, *happy university*⟩ might be translated into ⟨*?author*, *paper*⟩, linked by a *:writes* relation, and ⟨*?author*, *happy university*⟩, linked by a *:is_affiliated_with* relation). This abstract graph query could then be instantiated into a graph query language available for the graph database, where *?author* is a node of type *#author*. Despite the identified domain-dependent limitations of the model, this contributed to the application of graphs as a tool for natural language understanding and question answering.

Zhang et al. [148] explored graph-based document retrieval, by converting both documents and queries to graphs consisting of words and POS tags as nodes, and syntactic dependencies as edges. They segmented the documents into document semantic units (DSU), representing the atomic unit of parsing (e.g., a sentence, or a phrase within a sentence). They extracted graphs from each DSU, for each document. Node weights were computed from TF-IDF. They repeated this process for the query, considering it a single DSU. They then computed the maximum common subgraph between query and document graphs, taking into account that two nodes are the same if they share the same pair of word and POS tag. Node weights were combined based on the square root of the product, and edge weights were assigned based on whether the edge was present in both original graphs (1 if true and 0.5 if false). They calculated the similarity of a query graph and DSU graph based on a linear combination of the normalized sums of node and edge weights. The score for a document was computed based on the average of the similarities for all graphs representing document DSUs, or alternatively in a variant that assigned a higher weight to the title DSU. They prepared two datasets, one for Chinese and another one for English, by randomly selecting topics from the Sogou or Wikipedia top visited pages, and issuing those queries over Baidu or Google, in order to obtain result documents. Those documents were then graded for relevance by five human judges to form a test collection. Evaluation was done using DCG (discounted cumulative gain) and the best results were obtained with the title-biased score function. The graph-based approach outperformed the vector space model for both the Chinese and English test collections, and it even outperformed the Google algorithm.

[1] `https://dblp.org`

5.7 Hypergraph-based models

Hypergraphs [25] are a generalization of graphs, where edges (or hyperedges) can connect an arbitrary number of nodes — undirected hyperedges are represented by a set of nodes, while directed hyperedges are represented by a tuple of two sets of nodes. When all hyperedges in a hypergraph contain the same number k of nodes, the hypergraph is said to be k-uniform. In that case, it can be represented as a tensor of k dimensions, each of size $|V|$. In Section 5.5, we had covered tensor factorization over a tensor of entity relations for different predicates. Exploring analogous methods based on hypergraphs might also wield interesting results. A non-uniform hypergraph is called general. This is a family of hypergraphs that is rather hard to represent using tensors. CERN (Conseil Européen pour la Recherche Nucléaire) and the University of Geneva have recently been tackling this problem, focusing on undirected general hypergraphs [112], as well as hyperedges based on multisets instead of sets [113]. A hypergraph can also be called mixed [59][§4], when it contains both directed and undirected hyperedges, sometimes referred to as hyperarcs and hyperedges, respectively.

Hypergraphs have also been explored outside of mathematics. In information science, topic maps [64,146] have been used to jointly represent multiple indexes. Conceptually, topic maps are hypergraphs where nodes are topics or occurrences, and hyperedges are binary connections between topics and occurrences, or n-ary connections between topics. Garshol [64] has showed that topic maps can be used as a common reference model to represent metadata and subject-based classification, including controlled vocabularies, taxonomies, thesauri, faceted classification and ontologies. This means that, not only can topic maps be used to merge indexes, but also to extend the indexes with external knowledge, in order to improve search. While information retrieval was identified by Garshol as one of the main applications of topic maps, to this date not many actual applications can be found outside of information science. Yi [146] compared thesaurus-based information retrieval with topic-map-based information retrieval, by measuring the recall and search time of 40 participants over the two systems. He distinguished between queries based on a single concept (fact-based) and queries based on two or more concepts (relationship-based). They found that the topic-map-based system outperformed the thesaurus-based system, both regarding recall and search time, for relationship-based queries.

While hypergraphs have been previously used in information retrieval, they still don't play a major role in well-known tasks, despite their potential, as identified for instance in topic models. Perhaps the most notable work on hypergraphs for information retrieval is the query hypergraph proposed by Bendersky and Croft [24]. In the query hypergraph, nodes represent concepts from the query, and edges represent the dependencies between subsets of those nodes and a document. The query hypergraph is therefore able to represent higher-order term dependencies, capturing the "dependencies between term dependencies". Two types of hyperedges were defined: *local*, between individual concepts and the document; and *global*, between the entire set of concepts and the document. In order to obtain a score for a document and query, they relied on a factor graph representation of the hypergraph — a bipartite graph, where each hyperedge was represented by a factor node. The ranking function was then computed based on the local and global factors, that worked as document-dependent hyperedge weights. The approach is similar to other log-linear retrieval models, such as the Markov network model or the linear discrim-

inant model, however higher-order term dependencies are easier to incorporate into the model. Their methodical approach can be regarded as a fundamental step in supporting hypergraph-based work in information retrieval.

In entity-oriented search, we frequently deal with combined data or, at the very least, we separately work with corpora and knowledge bases. Accordingly, finding a joint representation for this kind of unstructured and structured data represents added value in the quest to reach general information retrieval. Menezes and Roth [100] have recently introduced semantic hypergraphs, proposing an approach to represent knowledge extracted from corpora based on recursive ordered hypergraphs. On one side, such extension of hypergraphs means that nodes, representing terms, can now have an order in the hyperedge they belong to, enabling for instance the representation of an entity mention to be stored using the correct sequence of words. On the other side, recursivity means that higher-order dependencies are explicitly stored rather than being exclusively verifiable, enabling a hyperedge to be defined over nodes but also over hyperedges. This work is also available as a Python library called Graphbrain[1], which can be used to manipulate semantic hypergraphs for natural language understanding, and knowledge inference and exploration.

Recently, Dietz [45] has also proposed ENT Rank, a hypergraph-based approach for entity ranking, where text was used to inform and improve entity retrieval. The hypergraph was then converted into an entity co-occurrence multigraph and several features were considered to train a learning-to-rank-entities model: neighbor features, relation-typed neighbor features, and context-relevance features. The model was inspired by random walks with restart, where training consisted on optimizing two weight vectors, $\vec{\psi}$ and $\vec{\theta}$, as part of an equation similar to PageRank's, which acts as the ranking function for a given entity. In this equation, the features for the scored entity corresponded to the teleport or restart term, while the features from the neighbors and context corresponded to the the navigation term. The author's evaluation was based on the entity retrieval task from the TREC Complex Answer Retrieval track. It relied on the CAR dataset, with 5.41 million Wikipedia pages, along with a large corpus of paragraphs with hyperlinks to Wikipedia pages. DBpedia-Entity v2 was also used, with relevance judgments from SemSearch ES, INEX-LD, List Search and QALD-2. ENT Rank was able to achieve first or second best ranking model for all experiments, showing, in multiple cases, the best performance for unsupervised ranked aggregation.

Assuming that we would be able to effectively represent text and entities using a hypergraph, then we might be able to take advantage of both set theory, using metrics like the Jaccard index to measure similarities, or random walks in hypergraphs [23], where we might use hyperedge weights, but also node weights to control the traversal, in order to support general information retrieval. While hypergraphs are a flexible data structure, they still present some limitations, when applied to more complex representation needs. For instance, weights associated with nodes and hyperedges might not be enough to represent all types of bias — e.g., we can define node weights, but not node weights per hyperedge. There are, however, extensions of hypergraphs, like fuzzy hypergraphs [89], intuitionistic fuzzy hypergraphs [3] or hypergraphs with edge-dependent vertex weights [39], that provide increased flexibility in establishing bias. According to Canfora and Cerulo [34][Fig.1], this would in fact mean that such a model would simultaneously provide reasoning with logic (graph theory) and

[1] `https://graphbrain.net`

uncertainty (fuzzy set theory). Besides hypergraphs, there are also other higher-order data structures, like higraphs [75], hypernetworks [2,84] or metagraphs [20], that might be worth exploring in information retrieval.

In prior sections of this survey, we have already seen that graphs can be used to represent both unstructured text (e.g., graph-of-word [126]) and structured knowledge (e.g., DBpedia [8]). Hypergraphs can go even further, capturing for instance synonyms as undirected hyperedges. Moreover, approaches like hypergraph embeddings [80] can also be used to further reduce search complexity. The expressiveness and viability of hypergraphs make it a useful data structure to be explored in entity-oriented search.

5.8 Random walk based models

Traversing a graph can be done through algorithms like breadth-first or depth-first search. For large graphs, however, the cost of using such strategies can be prohibitive. There are other less expensive traversal strategies, like random walks [96], that are still able to capture structural properties, but rely on a sampled view of the graph [90]. For example, while breadth-first search has time-complexity $O(|V|+|E|)$, a random walk has time-complexity $O(\ell)$, for a given length ℓ, while also being easily parallelizable [129]. Accordingly, random walks frequently provide a more efficient way to estimate network properties. They can be used for measuring node importance, when applied globally (e.g., *PageRank* [116]), but also for community detection, when confined to local neighborhoods (e.g., *Walktrap* [117], push algorithm [5]), and they can even be used for entity linking, when applied to graphs of mentions and entities [70][131][§3.2.4].

PageRank is perhaps the most well-known and versatile graph-based metric that relies on random walks. It first surfaced in 1997, in a working paper by Larry Page and Sergey Brin [115], but it is usually cited using the 1998 article describing the Google search engine [30], or the 1999 technical report from Stanford InfoLab [116]. Since then, PageRank has been extended and reimagined by different researchers, who proposed their own improvements, as we have shown in Section 5.1. Experiments included measuring the importance of web pages based on a given topic [76], or considering a weighted approach based on network, semantic and visual features [48], or even introducing higher-order dependencies for modeling historical surfing information [69]. There are multiple available surveys about PageRank, namely from Chung [41] and from Gleich [68]. Chung [41] focused on approximated approaches for the computation of PageRank, also covering the applications and generalization of PageRank. Gleich [68] provided an in-depth survey with a good coverage on existing PageRank variants and applications, discussing the mathematics of PageRank and its generalizations.

Due to its popularity, there are multiple applications of PageRank to entity-oriented search [12][§4.6.2]. In the remainder of this section, we present ReConRank, ObjectRank, HubRank and HopRank, with applications over RDF graphs or general labeled graphs. We present PopRank and DING, with applications over the semantic web, combining a web or dataset graph with an object or entity graph. Finally, we cover a semantics-aware personalized PageRank that explores PageRank for recommendation tasks, while considering RDF triples for improved performance.

ReConRank Inspired by PageRank, Hogan et al. [78] proposed ResourceRank, ContextRank, and a combination of the two approaches called ReConRank. Using a similar strategy to the base set selection in HITS [86], a query dependent graph was built by matching RDF literals and returning their neighborhood graph.
The resource graph was induced by the nodes that appeared at least once as a subject in the retrieved RDF quads, while the context graph was induced by the fourth elements in the same set of RDF quads. In practice, each graph represented different projections of the original graph. ReConRank was then proposed as a metric computed over the combined graph of resources and their contexts, which represented an enriched connectivity over either individual graph.

ObjectRank Balmin et al. [11] proposed an adaptation of PageRank for keyword search over a database modeled as a labeled graph. Like HITS, ObjectRank was computed over a base set to generate a topic-induced graph. Their approach consisted on precomputing the Global ObjectRank (same as PageRank) and the ObjectRanks for all term-based graphs.
For a query with multiple keywords, we can compute the product of individual ObjectRanks as the logical *AND* or, for pairs of keywords, the sum of ObjectRanks minus their product as the logical *OR*. It is also easy to derive the computation of ObjectRank for any combination of these boolean operators. A relevant difference between the computation of PageRank and ObjectRank, besides its query dependence according to the base set, is that the sum of outgoing weights, used to generate matrix $\mathcal{M}$, might be less than one. Weights are defined according to an authority transfer schema graph, where they are established for particular edge labels and between specific source and target node labels. Given that these weights might not add to one, the authors use the analogy of a random surfer that eventually disappears. For computational purposes, each weight is then divided by the weighted outdegree over edges with the same label, ensuring stochasticity and convergence.

PopRank Nie et al. [110] have proposed a link analysis metric with applications to entity ranking, namely in academic search engines like Libra [109] — know, since 2011, as Microsoft Academic Search. PopRank acknowledged the importance of both the web graph (based on hyperlinks) and the object graph (based on heterogeneous relations between different types of objects). It combines web popularity, as well as transitions over the object graph, according to a popularity propagation factor. The popularity propagation factor γ_{YX} was defined for links between two specific entity types Y and X (similar to the authority transfer schema graph in ObjectRank). The web popularity $WebPop_X$ was calculated based on the PageRank of the pages that contained the object, as well as based on the importance of web blocks (visual fragments of a web page).

HubRank Chakrabarti [36] proposed an efficiency improvement over ObjectRank, where the personalization vector was only computed for a set of hub nodes selected based on query logs. They proposed a *TypedWordGraph*, where they introduced word-to-entity relations, thus enabling mixed word and entity queries. Each vector was approximated using precomputed fingerprints — i.e., the end nodes from random walks of various lengths, as sampled from a geometric distribution, and initiated from each node — as described by Fogaras et al. [58]. In order to compute HubRank,

a subgraph limited by boundary nodes was first prepared. The boundary was established by a subset of hub nodes called blockers, and by loser nodes that were too far to significantly influence the personalized PageRank of the word nodes. Personalized PageRank was then estimated for the remaining active nodes and iteratively computed using dynamic programming, while fixing the value of boundary nodes. Fingerprinting and computation over a smaller graph provided improved efficiency, while the word-to-entity relations provided a more flexible model for entity-oriented search.

DING (Dataset rankING) Delbru et al. [44] proposed a hierarchical link analysis approach based on the computation of a PageRank variant called DatasetRank, applied over a two-layer model of the semantic web. DatasetRank combines a local entity rank, indicative of the importance of an entity within the current dataset, with the the probability of jumping to another dataset, which is dependent on its size.

Semantics-Aware Personalized PageRank Musto et al. [105] have experimented with personalized PageRank for recommendation over different user preference graphs, adding to the user-item relations with external knowledge from linked open data. Their contribution was focused on finding the best representation model for semantics-aware recommendation using personalized PageRank, rather than proposing changes to PageRank as a ranking function. They experimented with the bipartite user-item graph, as well as the tripartite user-item-resource graphs, based on all DBpedia triples, as well as on a subset of triples selected using PCA or information gain. They also experimented with different weighting schemes for each node type. They found slight benefits to the extension of user-item graphs with linked open data, particularly for graphs that were originally sparser.

HopRank Espín-Noboa et al. [51] proposed HopRank to model human navigation on semantic networks. Based on the analysis of user behavior in the BioPortal website[1], a repository of biomedical ontologies, they found that, instead of teleporting to random ontology nodes, users showed a bias toward jumping to nodes at a particular distance k. They called this a k-hop, naming the probabilities of teleporting to k-hops as *HopPortation*. Given the diameter d' of the ontology (ignoring direction), consider the *HopPortation* vector $\vec{d}$ of size $d'+1$, where $d_k \in \vec{d}$ represents the probability of a k-hop happening. The authors computed d_k based on the clickstream transitions in the BioPortal website, using add-one smoothing to ensure each available k-hop was considered. Also consider d' matrices $\mathcal{M}_k$ containing the transition probabilities for the corresponding k-hops, based on the undirected ontology links.

6 Discussion

In this section, we present several observations, identifying possible trails leading to the future of graph-based entity-oriented search. We end the section with an overview on the overall classes of graph-based models presented in this survey.

[1] `https://bioportal.bioontology.org/`

6.1 Observations

We present several remarks surrounding graph-based entity-oriented search, its relation to semantic search and the exploration of higher-order dependencies with hypergraphs, proposing future directions towards hypergraph-based quantum search[1].

6.1.1 Why survey graph-based entity-oriented search?

Only recently has entity-oriented search been conveniently defined and described as an area [12]. While several graph-based approaches have been generally used in information retrieval, graph-based entity-oriented search is still in its infancy. It lies within this area the ability to tackle issues like the combination of heterogeneous information sources or the generalization of entity-oriented search tasks — all available information, structured or unstructured, should be available for cross-referencing, collectively contributing to solving the users' information needs. Likewise, individual tasks leading to the answer might benefit from a departure from modularity and into a more intertwined approach, where intermediate computations from any task should be able to contribute to other tasks, seamlessly at any step. In order to develop such a holistic approach to entity-oriented search, we must first compile a comprehensive guide with a high-level view over information retrieval, and in particular the developments leading to entity-oriented search and the overall usage of graphs in the area. Our goal with this survey was to solve for this need, striving to be complete in the sense of coverage, as opposed to being exhaustive, and showing the potential for tackling information retrieval as the analysis of a complex network.

6.1.2 The relation between entity-oriented search and semantic search

One particular source of confusion is the definition of semantic search and how it relates to entity-oriented search. Most of the work we reviewed either refers to semantic search as document retrieval leveraging entities, or as entity retrieval over linked data. In its broader definition [12][Def.1.6], semantic search subsumes entity-oriented search. However, when considering any of the described tasks, we might say that semantic search is instead subsumed by entity-oriented search. In practice, detaching the semantic search classification from any specific task might be the most adequate approach, thus promoting the use of the broader and more abstract definition, and instead more clearly describing the tasks as ad hoc document retrieval and ad hoc entity retrieval, respectively. In this survey, we complied with the definitions proposed by Balog [12], except when the cited paper specifically mentioned a semantic search task, in which case we clearly stated which definition the authors adhered to.

6.1.3 What is and isn't a graph-based model?

We defined graph-based models as any approach that relied on a graph, at whichever stage of the process. This included graphs for representing:

[1] Please note that quantum approaches to information retrieval have already been explored in the past, for instance with the quantum language models by Sordoni et al. [134]

- Text (e.g., linking terms within a window, or with similar embeddings);
- Entities, their attributes and relations (i.e., knowledge graphs);
- Relations between documents (e.g., hyperlinks, similarity).

While many probabilistic models might also be considered graph-based, namely Bayesian or Markov networks, we opted to classify them as probabilistic, unless they were clearly operating over a specific graph (e.g., web graph, similarity graph). PageRank might be the most evident example of a probabilistic graph-based model, since it is clearly applied to the web graph to rank web pages by importance. This is why it was relevant to cover overall probabilistic models, before delving into graph-based models.

As an area, graph-based entity-oriented search still has a lot of unexploited potential, in particular regarding approaches developed in network science. This includes PageRank, which as been abundantly used, but also other centrality metrics like closeness or betweenness, as well as community detection or motif discovery. Graph connectivity can be studied from three main perspectives: microscale (node or edge properties), mesoscale (community or motif level) and macroscale (global). There are still many unexplored approaches, at all scales, that might be useful to better understand information in the context of search (e.g., graphlet-orbit transitions as a way to establish graph similarity [6]).

6.1.4 From binary dependencies to higher-order dependencies

A current trend in machine learning is the application of tensors for representing higher-order dependencies, particularly popularized by Google's TensorFlow [1]. Similarly, hypergraphs are able to elevate the expressiveness of a graph's binary dependencies to higher-order dependencies. In Section 5.7, we have seen that there some hypergraph-based approaches for indexing, representing and querying documents. However, there hasn't been much work specifically directed at entity-oriented search. We argue that further exploring hypergraphs, without falling back to the domain of graphs, might lead to useful and novel strategies to better solve information needs. A possible approach is the application of PageRank to "knowledge hypergraphs", where a random surfer would, at each step, randomly select a hyperedge and then randomly select a node from that hyperedge [23]. As the complexity of the hypergraph increases, particularly for memory-based hypergraphs (i.e., that explicitly store information statements), even random walk based approaches become inefficient for real-time computation. However, we know that random walks in graphs can be modeled using Markov chains, which are stochastic models whose simulation is being studied in quantum computer [111]. In turn, implementing random walks in hypergraphs using a quantum computer would also require a Markov process to be defined over a hypergraph [95]. We also argue that, for this reason, the complexity of such models and the overall predicted inefficiency should not be reasons to discard it as a viable approach, worthy of study.

6.2 An overview on entity-oriented search approaches

Entity-oriented search is a naturally heterogeneous area, where documents and entities are combined to better solve the information needs of the users. When querying,

users can take advantage of keyword or natural language queries, as well as entity queries, obtaining results that can either include documents, entities, or both. However, techniques for document and entity representation have been quite disjoint, with the inverted index taking the lead to represent multi-field documents, and the triplestore taking the lead to represent entities, their types, attributes and relations. Some of the first approaches to tackling entity-oriented search tasks, were based on translating the problem to the domain of classical information retrieval. Please refer to Table 2 for an overview of these approaches, based on virtual documents, combined data and probabilistic graphical models. Other approaches integrated information from documents and entities based on learning to rank models. That way, signals from different representations (e.g., inverted index and triplestore) could be combined based on a learned ranking function, trained for instance using a support-vector machine or a neural network. Table 3 can be used as a reference for the learning to rank models that we covered, illustrating semantic-driven, virtual document, and representation learning approaches.

Graph-based models can also be used as a way to integrate information from documents and entities, harnessing techniques developed through years of research on information retrieval and network science (e.g., PageRank [116]), as well as graph-based representations developed individually for either type of data (e.g., graph-of-word [126] for documents and RDF[1] for entities). While graphs have been prevalent in information retrieval, using them to solve the representation mismatch between documents and entities is fairly recent. Table 4 provides a reference for graph-based approaches, both general and specific to entity-oriented search. In particular, we covered general link analysis approaches. We also covered text as a graph, which, despite no entities being considered, provided a common ground for integration with entity graphs. We then covered knowledge graphs and how they are built and used for document and/or entity search. We examined text to entity graph approaches, where information extraction was used to acquire a structured graph to represent the document by its entities and relations. We then covered graph matching approaches, where a query graph is matched against subgraphs in an entity graph. Finally, we considered hypergraph-based models, with potential applications to entity-oriented search, and we closed with random walk based models, that are based on PageRank adaptions to an entity-oriented context.

Table 1 provides a comprehensive view of the surveyed approaches for each of the three models — classical IR, learning to rank, and graph-based models — along with the tasks that they support.

As we can see, the task with the highest coverage was ad hoc entity retrieval. Combined data approaches are able to support all of the four main entity-oriented search tasks that we considered. The reviewed graph matching and random walk based approaches are able to support three out of the four tasks, with ad hoc document retrieval missing. However, graph-based models were used to represent text as a graph, to structure knowledge bases, to convert text to an entity graph, and in hypergraph-based approaches for ad hoc document retrieval. This supports our thesis that the graph data structure might be viable as a joint representation model, able to support the four retrieval tasks, and providing a framework to develop a universal ranking function. One example of a basis for such a function would be the heat

[1] `https://www.w3.org/RDF/`

Table 1: Approaches and their applications in the context of entity-oriented search.

	Approach	References	Ad Hoc Document Retrieval	Ad Hoc Entity Retrieval	Related Entity Finding	Entity List Completion	Sentence Retrieval	Answer Tree Ranking	Attribute Retrieval	Relation Retrieval	Knowledge Graph Construction and Modeling	Node Importance	Node Relatedness	Graph Partitioning	Topic Modeling	Text Classification	Joint Representation	Document Representation
Classical	Virtual Documents	[21, 47, 118]		✓					✓	✓								
	Combined Data	[26, 18, 152, 31]	✓	✓	✓	✓											✓	
	Probabilistic Graphical Models	[88, 120, 139]		✓			✓											
	Cluster Hypothesis	[121]	✓															
Learning to Rank	Semantic-Driven	[38, 92, 130]		✓	✓		✓											
	Virtual Documents	[37]		✓														
	Representation Learning	[73]		✓														
Graph-Based	Link Analysis	[86, 30, 116, 140, 82, 40, 87, 144]										✓	✓	✓				
	Text as a Graph	[27, 126, 49]	✓													✓		✓
	Knowledge Graphs	[55, 33, 13, 28, 106, 133, 119, 63, 4]	✓	✓							✓				✓		✓	
	Text to Entity Graph	[29, 107]	✓	✓														
	Entity Graph to Tensor	[149]		✓														
	Graph Matching	[153, 150, 154, 103, 151]		✓	✓	✓		✓										
	Hypergraph-Based	[64, 146, 24, 74, 45]	✓	✓													✓	✓
	Random Walk Based	[78, 11, 36, 51, 44, 105]		✓	✓	✓						✓						

kernel PageRank, which is able to measure node importance and node relatedness, as well as a to obtain a graph partition.

7 Conclusion

With the increasing relevance of entity-oriented search, it makes sense to look at graph-based models for information retrieval in a new light. We started this survey by providing context, presenting some historical perspective along with basic concepts and models from information retrieval. We covered general entity-oriented search approaches and general graph-based approaches, as well as a combination of both approaches. Given the growing potential of the area, this survey focused on identifying a diverse set of representative methods, rather than doing an exhaustive research of all existing applications in each category. Our goal was to provide a map of opportunities in graph-based entity-oriented search, supporting, among others, the future research on general models and universal ranking functions for information retrieval.

We surveyed the usage of classical information retrieval models, as well as learning to rank models, for entity-oriented search. Then, we provided a wide range coverage of graph-based models, introducing classical link analysis approaches, like PageRank, HITS and kernel-based methods. We also described approaches for representing text as a graph, capturing discourse properties like context (e.g., graph-of-word). We

described knowledge graph construction and modeling, along with its applications, either for improving ad hoc document retrieval or for supporting ad hoc entity retrieval. We studied approaches based on extracting entity graphs from text and using them as a complement for the representation and retrieval of documents. We also covered the usage of tensors to represent entity graphs and to obtain entity embeddings. We explored graph matching for querying with graphs — usually generated from natural language queries. We examined general hypergraph-based models for document representation, joint representation and ad hoc document retrieval, showing the potential for applications in entity-oriented search as well. We closed the graph-based section with random walk based models, mostly derived from PageRank and applied to entity graphs over a given context (e.g., web graph, dataset). We also provided a section on evaluation forums and datasets, useful for assessing a wide range of entity-oriented search tasks. Finally, we presented a discussion containing several individual observations, as well as an overview on the surveyed entity-oriented search approaches.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283. USENIX Association, Savannah, GA (2016). URL `https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi`
2. Akhmediyarova, A., Kuandykova, J., Kubekov, B., Utepbergenov, I.T., Popkov, V.: Objective of modeling and computation of city electric transportation networks properties. In: Proc. of the Int. Conf. on Information Science and Management Engineering, Destech Publications, Inc., pp. 106–111 (2015)
3. Akram, M., Dudek, W.A.: Intuitionistic fuzzy hypergraphs with applications. Inf. Sci. **218**, 182–193 (2013). DOI 10.1016/j.ins.2012.06.024
4. Allahyari, M.: Semantic web topic models: Integrating ontological knowledge and probabilistic topic models. Ph.D. thesis, University of Georgia, Athens, Georgia (2016)
5. Andersen, R., Chung, F.R.K., Lang, K.J.: Local graph partitioning using pagerank vectors. In: 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings, pp. 475–486 (2006). DOI 10.1109/FOCS.2006.44
6. Aparicio, D., Ribeiro, P., Silva, F.: Graphlet-orbit transitions (got): A fingerprint for temporal network comparison. PLoS One **13**, e0205497 (2018). DOI 10.1371/journal.pone.0205497
7. Arrington, M.: AOL proudly releases massive amounts of private data. `https://techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/` (2006). Accessed on 2017-07-13
8. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007., pp. 722–735 (2007). DOI 10.1007/978-3-540-76298-0_52
9. Avrachenkov, K., Litvak, N., Nemirovsky, D., Osipova, N.: Monte carlo methods in PageRank computation: When one iteration is sufficient. SIAM J. Numerical Analysis **45**(2), 890–904 (2007). DOI 10.1137/050643799

10. Baeza-Yates, R., Ciaramita, M., Mika, P., Zaragoza, H.: Towards semantic search. In: E. Kapetanios, V. Sugumaran, M. Spiliopoulou (eds.) Natural Language and Information Systems, 13th International Conference on Applications of Natural Language to Information Systems, NLDB 2008, London, UK, June 24-27, 2008, Proceedings, *Lecture Notes in Computer Science*, vol. 5039, pp. 4–11. Springer (2008). DOI 10.1007/978-3-540-69858-6_2
11. Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: Authority-based keyword search in databases. In: (e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004, pp. 564–575 (2004). URL `http://www.vldb.org/conf/2004/RS15P2.PDF`
12. Balog, K.: Entity-Oriented Search, *The Information Retrieval Series*, vol. 39. Springer (2018). DOI 10.1007/978-3-319-93935-3
13. Balog, K., de Rijke, M., Franz, R., Peetz, H., Brinkman, B., Johgi, I., Hirschel, M.: SaHaRa: Discovering Entity-Topic Associations in Online News. In: 8th International Semantic Web Conference (ISWC 2009) (2009)
14. Balog, K., Serdyukov, P., de Vries, A.P.: Overview of the TREC 2011 entity track. In: Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011 (2011). URL `http://trec.nist.gov/pubs/trec20/papers/ENTITY.OVERVIEW.pdf`
15. Bar-Yossef, Z., Mashiach, L.: Local approximation of pagerank and reverse pagerank. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008, pp. 865–866 (2008). DOI 10.1145/1390334.1390545
16. Baraglia, R., De Francisci Morales, G., Lucchese, C.: Document similarity self-join with MapReduce. In: 2010 IEEE 10th International Conference on Data Mining (ICDM 2010), pp. 731–736 (2010). DOI 10.1109/ICDM.2010.70
17. Bast, H., Bäurle, F., Buchhold, B., Haussmann, E.: Broccoli: Semantic full-text search at your fingertips. CoRR **abs/1207.2615** (2012). URL `http://arxiv.org/abs/1207.2615`
18. Bast, H., Buchhold, B.: An Index for Efficient Semantic Full-text Search. In: Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management, pp. 369–378 (2013). DOI 10.1145/2505515.2505689
19. Bast, H., Buchhold, B., Haussmann, E.: Semantic search on text and knowledge bases. Found. Trends Inf. Retr. **10**(2-3), 119–271 (2016). DOI 10.1561/1500000032
20. Basu, A., Blanning, R.W.: Metagraphs: A tool for modeling decision support systems. Management Science **40**(12), 1579–1600 (1994). URL `https://www.jstor.org/stable/2632940`
21. Bautin, M., Skiena, S.: Concordance-based entity-oriented search. In: 2007 IEEE / WIC / ACM International Conference on Web Intelligence, WI 2007, 2-5 November 2007, Silicon Valley, CA, USA, Main Conference Proceedings, pp. 586–592. IEEE Computer Society (2007). DOI 10.1109/WI.2007.84
22. Bavelas, A.: Communication patterns in task-oriented groups. The Journal of the Acoustical Society of America **22**(6), 725–730 (1950). DOI 10.1121/1.1906679
23. Bellaachia, A., Al-Dhelaan, M.: Random walks in hypergraph. In: Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods, Venice Italy, pp. 187–194 (2013). URL `http://www.inase.org/library/2013/venice/bypaper/AMCM/AMCM-28.pdf`
24. Bendersky, M., Croft, W.B.: Modeling higher-order term dependencies in information retrieval using query hypergraphs. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, pp. 941–950. ACM, New York, NY, USA (2012). DOI 10.1145/2348283.2348408
25. Berge, C.: Graphes et hypergraphes. Dunod: Paris (1970)
26. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid search: Effectively combining keywords and semantic searches. In: S. Bechhofer, M. Hauswirth, J. Hoffmann, M. Koubarakis (eds.) The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings, *Lecture Notes in Computer Science*, vol. 5021, pp. 554–568. Springer (2008). DOI 10.1007/978-3-540-68234-9_41
27. Blanco, R., Lioma, C.: Graph-based term weighting for information retrieval. Information Retrieval **15**(1), 54–92 (2012). DOI 10.1007/s10791-011-9172-x
28. Blanco, R., Mika, P., Vigna, S.: Effective and efficient entity search in RDF data. In: The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn,

Germany, October 23-27, 2011, Proceedings, Part I, pp. 83–97 (2011). DOI 10.1007/978-3-642-25073-6_6
29. Bordino, I., Mejova, Y., Lalmas, M.: Penguins in sweaters, or serendipitous entity search on user-generated content. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013), pp. 109–118 (2013). DOI 10.1145/2505515.2505680
30. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Networks **30**(1-7), 107–117 (1998). DOI 10.1016/S0169-7552(98)00110-X
31. Bron, M., Balog, K., de Rijke, M.: Example based entity search in the web of data. In: Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings, pp. 392–403 (2013). DOI 10.1007/978-3-642-36973-5_33
32. Bu, J., Tan, S., Chen, C., Wang, C., Wu, H., Zhang, L., He, X.: Music recommendation by unified hypergraph: combining social media information and music content. In: Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010, pp. 391–400 (2010). DOI 10.1145/1873951.1874005
33. Byrne, K.: Populating the semantic web — combining text and relational databases as rdf graphs. Ph.D. thesis, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh (2009). URL `http://hdl.handle.net/1842/3781`
34. Canfora, G., Cerulo, L.: A taxonomy of information retrieval models and tools. Journal of Computing and Information Technology **12**(3), 175–194 (2004). DOI 10.2498/cit.2004.03.01
35. Cattuto, C., Schmitz, C., Baldassarri, A., Servedio, V.D.P., Loreto, V., Hotho, A., Grahl, M., Stumme, G.: Network properties of folksonomies. AI Commun. **20**(4), 245–262 (2007). URL `http://content.iospress.com/articles/ai-communications/aic410`
36. Chakrabarti, S.: Dynamic personalized PageRank in entity-relation graphs. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007, pp. 571–580 (2007). DOI 10.1145/1242572.1242650
37. Chen, J., Xiong, C., Callan, J.: An empirical study of learning to rank for entity search. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016, pp. 737–740 (2016). DOI 10.1145/2911451.2914725
38. Chen, R., Spina, D., Croft, W.B., Sanderson, M., Scholer, F.: Harnessing semantics for answer sentence retrieval. In: K. Balog, J. Dalton, A. Doucet, Y. Ibrahim (eds.) Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, ESAIR 2015, Melbourne, Australia, October 23, 2015, pp. 21–27. ACM (2015). DOI 10.1145/2810133.2810136
39. Chitra, U., Raphael, B.J.: Random walks on hypergraphs with edge-dependent vertex weights. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pp. 1172–1181 (2019). URL `http://proceedings.mlr.press/v97/chitra19a.html`
40. Chung, F.: The heat kernel as the pagerank of a graph. Proceedings of the National Academy of Sciences **104**(50), 19735–19740 (2007). DOI 10.1073/pnas.0708838104
41. Chung, F.: A brief survey of PageRank algorithms. IEEE Trans. Network Science and Engineering **1**(1), 38–42 (2014). DOI 10.1109/TNSE.2014.2380315
42. Corso, G.M.D., Gulli, A., Romani, F.: Fast pagerank computation via a sparse linear system. Internet Mathematics **2**(3), 251–273 (2005). DOI 10.1080/15427951.2005.10129108
43. Craswell, N., Robertson, S.E., Zaragoza, H., Taylor, M.J.: Relevance weighting for query independent evidence. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005, pp. 416–423 (2005). DOI 10.1145/1076034.1076106
44. Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., Decker, S.: Hierarchical link analysis for ranking web data. In: The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part II, pp. 225–239 (2010). DOI 10.1007/978-3-642-13489-0_16
45. Dietz, L.: ENT rank: Retrieving entities for topical information needs through entity-neighbor-text relations. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, pp. 215–224 (2019). DOI 10.1145/3331184.3331257

46. Dietz, L., Schuhmacher, M.: An interface sketch for queripidia: Query-driven knowledge portfolios from the web. In: K. Balog, J. Dalton, A. Doucet, Y. Ibrahim (eds.) Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, ESAIR 2015, Melbourne, Australia, October 23, 2015, pp. 43–46. ACM (2015). DOI 10.1145/2810133.2810145
47. Dietz, L., Schuhmacher, M., Ponzetto, S.P.: Queripidia: Query-specific wikipedia construction. Proceedings of the 4th Workshop on Automated Knowledge Base Construction (AKBC 2014) (2014). URL `http://ciir-publications.cs.umass.edu/pub/web/getpdf.php?id=1174`
48. Dimitrov, D., Singer, P., Lemmerich, F., Strohmaier, M.: What makes a link successful on wikipedia? In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017, pp. 917–926 (2017). DOI 10.1145/3038912.3052613
49. Dourado, Í.C., Galante, R., Gonçalves, M.A., da Silva Torres, R.: Bag of textual graphs (botg): A general graph-based text representation model. J. Assoc. Inf. Sci. Technol. **70**(8), 817–829 (2019). DOI 10.1002/asi.24167
50. Emtage, A., Deutsch, P.: Archie: An electronic directory service for the internet. In: Proceedings of the USENIX Winter 1992 Technical Conference, pp. 93–110. San Francisco, CA, USA (1992)
51. Espín-Noboa, L., Lemmerich, F., Walk, S., Strohmaier, M., Musen, M.A.: Hoprank: How semantic structure influences teleportation in pagerank (A case study on bioportal). In: The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, pp. 2708–2714 (2019). DOI 10.1145/3308558.3313487
52. Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k lists. SIAM J. Discrete Math. **17**(1), 134–160 (2003). URL `http://epubs.siam.org/sam-bin/dbq/article/41285`
53. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004, pp. 49–56 (2004). DOI 10.1145/1008992.1009004
54. Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E.: Semantically enhanced information retrieval: An ontology-based approach. J. Web Semant. **9**(4), 434–452 (2011). DOI 10.1016/j.websem.2010.11.003
55. Fernández, M., López, V., Sabou, M., Uren, V.S., Vallet, D., Motta, E., Castells, P.: Semantic search meets the web. In: Proceedings of the 2th IEEE International Conference on Semantic Computing (ICSC 2008), August 4-7, 2008, Santa Clara, California, USA, pp. 253–260. IEEE Computer Society (2008). DOI 10.1109/ICSC.2008.52
56. Fletcher, G.H.L., Hidders, J., Larriba-Pey, J.L. (eds.): Graph Data Management, Fundamental Issues and Recent Developments. Data-Centric Systems and Applications. Springer (2018). DOI 10.1007/978-3-319-96193-4
57. Fogaras, D.: Where to start browsing the web? In: Innovative Internet Community Systems, Third International Workshop, IICS 2003, Leipzig, Germany, June 19-21, 2003, Revised Papers, pp. 65–79 (2003). DOI 10.1007/978-3-540-39884-4_6
58. Fogaras, D., Rácz, B., Csalogány, K., Sarlós, T.: Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments. Internet Mathematics **2**(3), 333–358 (2005). DOI 10.1080/15427951.2005.10129104
59. Frank, A., Király, T., Király, Z.: On the orientation of graphs and hypergraphs. Discrete Applied Mathematics **131**(2), 385–400 (2003). DOI 10.1016/S0166-218X(02)00462-6
60. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry **40**(1), 35 (1977). DOI 10.2307/3033543
61. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: M.M. Veloso (ed.) IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, pp. 1606–1611 (2007). URL `http://ijcai.org/Proceedings/07/Papers/259.pdf`
62. Ganea, O., Hofmann, T.: Deep joint entity disambiguation with local neural attention. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pp. 2619–2629 (2017). URL `https://aclanthology.info/papers/D17-1277/d17-1277`
63. Gao, Y., Liang, J., Han, B., Yakout, M., Mohamed, A.: KDD tutorial T39: Building a large-scale, accurate and fresh knowledge graph. `https://kdd2018tutorialt39.azurewebsites.net/` (2018). Accessed on 2019-05-16

64. Garshol, L.M.: Metadata? thesauri? taxonomies? topic maps! making sense of it all. J. Information Science **30**(4), 378–391 (2004). DOI 10.1177/0165551504045856
65. Gerritse, E.J., Hasibi, F., de Vries, A.P.: Graph-embedding empowered entity retrieval. In: J.M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M.J. Silva, F. Martins (eds.) Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I, *Lecture Notes in Computer Science*, vol. 12035, pp. 97–110. Springer (2020). DOI 10.1007/978-3-030-45439-5_7. URL https://doi.org/10.1007/978-3-030-45439-5_7
66. Getoor, L., Diehl, C.P.: Link mining: A survey. SIGKDD Explor. Newsl. **7**(2), 3–12 (2005). DOI 10.1145/1117454.1117456
67. Gleich, D., Zhukov, L., Berkhin, P.: Fast parallel PageRank: A linear system approach. Tech. Rep. YRL-2004-038, Yahoo! Research (2004). URL http://research.yahoo.com/publication/YRL-2004-038.pdf
68. Gleich, D.F.: Pagerank beyond the web. SIAM Review **57**(3), 321–363 (2015). DOI 10.1137/140976649
69. Gleich, D.F., Lim, L., Yu, Y.: Multilinear PageRank. SIAM J. Matrix Analysis Applications **36**(4), 1507–1541 (2015). DOI 10.1137/140985160
70. Guo, Z., Barbosa, D.: Robust entity linking via random walks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, pp. 499–508 (2014). DOI 10.1145/2661829.2661887
71. Gupta, M., Bendersky, M.: Information retrieval with verbose queries. Foundations and Trends in Information Retrieval **9**(3-4), 91–208 (2015). DOI 10.1561/1500000050
72. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.O.: Combating web spam with trustrank. In: (e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004, pp. 576–587 (2004). URL http://www.vldb.org/conf/2004/RS15P3.PDF
73. Gysel, C.V., de Rijke, M., Kanoulas, E.: Learning latent vector spaces for product search. In: Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016, pp. 165–174 (2016). DOI 10.1145/2983323.2983702
74. Haentjens Dekker, R., Birnbaum, D.J.: It's more than just overlap: Text As Graph. In: Proceedings of Balisage: The Markup Conference 2017, vol. 19 (2017). DOI 10.4242/BalisageVol19.Dekker01
75. Harel, D.: On visual formalisms. Commun. ACM **31**(5), 514–530 (1988). DOI 10.1145/42411.42414
76. Haveliwala, T.H.: Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. IEEE Trans. Knowl. Data Eng. **15**(4), 784–796 (2003). DOI 10.1109/TKDE.2003.1208999
77. Herrera, J., Hogan, A., Käfer, T.: BTC-2019: the 2019 billion triple challenge dataset. In: The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II, pp. 163–180 (2019). DOI 10.1007/978-3-030-30796-7_11
78. Hogan, A., Harth, A., Decker, S.: ReConRank: A scalable ranking method for semantic web data with context. In: Proceedings of Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006), in conjunction with International Semantic Web Conference (ISWC 2006) (2006). URL http://hdl.handle.net/10379/492
79. Huang, A., Milne, D.N., Frank, E., Witten, I.H.: Learning a concept-based document similarity measure. Journal of the Association for Information Science and Technology **63**(8), 1593–1608 (2012). DOI 10.1002/asi.22689
80. Huang, J., Chen, C., Ye, F., Wu, J., Zheng, Z., Ling, G.: Hyper2vec: Biased random walk for hyper-network embedding. In: Database Systems for Advanced Applications - DASFAA 2019 International Workshops: BDMS, BDQM, and GDMA, Chiang Mai, Thailand, April 22-25, 2019, Proceedings, pp. 273–277 (2019). DOI 10.1007/978-3-030-18590-9_27
81. Irrera, O., Silvello, G.: Background linking: Joining entity linking with learning to rank models. In: D. Dosso, S. Ferilli, P. Manghi, A. Poggi, G. Serra, G. Silvello (eds.) Proceedings of the 17th Italian Research Conference on Digital Libraries, Padua, Italy (virtual event due to the Covid-19 pandemic), February 18-19, 2021, *CEUR Workshop Proceedings*, vol. 2816, pp. 64–77. CEUR-WS.org (2021). URL http://ceur-ws.org/Vol-2816/paper6.pdf

82. Ito, T., Shimbo, M., Kudo, T., Matsumoto, Y.: Application of kernels to link analysis. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005, pp. 586–592 (2005). DOI 10.1145/1081870.1081941
83. Jespersen, O.: The philosophy of grammar. Routledge (2013 [1924]). URL 10.4324/9780203716045
84. Johnson, J.: Hypernetworks in the Science of Complex Systems, *Series on Complexity Science*, vol. 3. World Scientific (2014). DOI 10.1142/p533
85. Kandola, J.S., Shawe-Taylor, J., Cristianini, N.: Learning semantic similarity. In: Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada], pp. 657–664 (2002). URL http://papers.nips.cc/paper/2316-learning-semantic-similarity
86. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999). DOI 10.1145/324133.324140
87. Kloster, K., Gleich, D.F.: Heat kernel based community detection. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, pp. 1386–1395 (2014). DOI 10.1145/2623330.2623706
88. Koumenides, C.L., Shadbolt, N.R.: Combining link and content-based information in a Bayesian inference model for entity search. In: Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search - JIWES '12, pp. 1–6 (2012). DOI 10.1145/2379307.2379310
89. Lee-Kwang, H., Lee, K.: Fuzzy hypergraph and fuzzy partition. IEEE Trans. Systems, Man, and Cybernetics **25**(1), 196–201 (1995). DOI 10.1109/21.362951
90. Leskovec, J., Faloutsos, C.: Sampling from large graphs. In: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006, pp. 631–636 (2006). DOI 10.1145/1150402.1150479
91. Li, J., Zhang, L., Yu, Y.: Learning to generate semantic annotation for domain specific sentences. In: Proceedings of the K-CAP 2001 Workshop on Knowledge Markup and Semantic Annotation Victoria, B.C., Canada, October 21, 2001 (2001). URL http://ceur-ws.org/Vol-99/Jianming_Li-et-al.pdf
92. Lin, B., Rosa, K.D., Shah, R., Agarwal, N.: LADS: Rapid development of a learning-to-rank based related entity finding system using open advancement. In: Proceedings of The First International Workshop on Entity-Oriented Search (EOS) (2011)
93. Lloyd, L., Kechagias, D., Skiena, S.: Lydia: A system for large-scale news analysis. In: M.P. Consens, G. Navarro (eds.) String Processing and Information Retrieval, 12th International Conference, SPIRE 2005, Buenos Aires, Argentina, November 2-4, 2005, Proceedings, *Lecture Notes in Computer Science*, vol. 3772, pp. 161–166. Springer (2005). DOI 10.1007/11575832_18
94. López, V., Sabou, M., Motta, E.: Powermap: Mapping the real semantic web on the fly. In: The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings, pp. 414–427 (2006). DOI 10.1007/11926078_30
95. Louis, A.: Hypergraph markov operators, eigenvalues and approximation algorithms. In: Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015, pp. 713–722 (2015). DOI 10.1145/2746539.2746555
96. Lovász, L., et al.: Random walks on graphs: A survey. Combinatorics, Paul erdos is eighty **2**(1), 1–46 (1993)
97. Lv, Y., Zhai, C.: Lower-bounding term frequency normalization. In: Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011, pp. 7–16 (2011). DOI 10.1145/2063576.2063584
98. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press (2008). DOI 10.1017/CBO9780511809071. URL https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf
99. McFee, B., Lanckriet, G.R.G.: Hypergraph models of playlist dialects. In: Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012, pp. 343–348 (2012). URL http://ismir2012.ismir.net/event/papers/343-ismir-2012.pdf

100. Menezes, T., Roth, C.: Semantic hypergraphs. CoRR **abs/1908.10784** (2019). URL http://arxiv.org/abs/1908.10784
101. Metzler, D., Kanungo, T.: Machine learned sentence selection strategies for query-biased summarization. In: Proceedings of SIGIR 2008 Workshop on Learning to Rank for Information Retrieval (LR4IR), held in conjunction with the 31th Annual International ACM SIGIR Conference, pp. 40–47. Singapore (2008)
102. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pp. 3111–3119 (2013). URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality
103. Minkov, E., Cohen, W.W.: Improving graph-walk-based similarity with reranking: Case studies for personal information management. ACM Trans. Inf. Syst. **29**(1), 4:1–4:52 (2010). DOI 10.1145/1877766.1877770
104. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Trans. Assoc. Comput. Linguistics **2**, 231–244 (2014). URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/291
105. Musto, C., Semeraro, G., de Gemmis, M., Lops, P.: Tuning personalized PageRank for semantics-aware recommendations based on linked open data. In: The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I, pp. 169–183 (2017). DOI 10.1007/978-3-319-58068-5_11
106. Neumayer, R., Balog, K., Nørvåg, K.: On the modeling of entities for ad-hoc entity search in the web of data. In: Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012. Proceedings, pp. 133–145 (2012). DOI 10.1007/978-3-642-28997-2_12
107. Ni, Y., Xu, Q.K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H.J., Cao, S.S.: Semantic Documents Relatedness using Concept Graph Representation. In: Proceedings of the 9th ACM International Conference on Web Search and Data Mining - WSDM '16, pp. 635–644. ACM Press, New York, New York, USA (2016). DOI 10.1145/2835776.2835801
108. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pp. 809–816 (2011). URL https://icml.cc/2011/papers/438_icmlpaper.pdf
109. Nie, Z., Wen, J., Ma, W.: Object-level vertical search. In: CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings, pp. 235–246 (2007). URL http://cidrdb.org/cidr2007/papers/cidr07p26.pdf
110. Nie, Z., Zhang, Y., Wen, J., Ma, W.: Object-level ranking: bringing order to web objects. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005, pp. 567–574 (2005). DOI 10.1145/1060745.1060828
111. Nikolov, P., Galabov, V.: Markov process simulation on a real quantum computer. Proceedings of the 45th International Conference on Application of Mathematics in Engineering and Economics (AMEE 2019) (2019). DOI 10.1063/1.5133584
112. Ouvrard, X., Goff, J.L., Marchand-Maillet, S.: Adjacency and tensor representation in general hypergraphs part 1: e-adjacency tensor uniformisation using homogeneous polynomials. CoRR **abs/1712.08189** (2017)
113. Ouvrard, X., Goff, J.L., Marchand-Maillet, S.: Adjacency and tensor representation in general hypergraphs.part 2: Multisets, hb-graphs and related e-adjacency tensors. CoRR **abs/1805.11952** (2018)
114. Oza, P., Dietz, L.: Which entities are relevant for the story? In: R. Campos, A.M. Jorge, A. Jatowt, S. Bhatia, M.A. Finlayson (eds.) Proceedings of Text2Story - Fourth Workshop on Narrative Extraction From Texts held in conjunction with the 43rd European Conference on Information Retrieval (ECIR 2021), Lucca, Italy, April 1, 2021 (online event due to Covid-19 outbreak), *CEUR Workshop Proceedings*, vol. 2860, pp. 41–48. CEUR-WS.org (2021). URL http://ceur-ws.org/Vol-2860/paper5.pdf
115. Page, L.: PageRank: Bringing order to the web. Tech. rep., Stanford Digital Libraries Working Paper (1997). URL http://www-diglib.stanford.edu/diglib/WP/PUBLIC/DOC159.html
116. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999). URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120

117. Pons, P., Latapy, M.: Computing communities in large networks using random walks. Journal of Graph Algorithms and Applications **10**(2), 191–218 (2006). URL http://jgaa.info/accepted/2006/PonsLatapy2006.10.2.pdf
118. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: M. Rappa, P. Jones, J. Freire, S. Chakrabarti (eds.) Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010, pp. 771–780. ACM (2010). DOI 10.1145/1772690.1772769
119. Qian, R.: Bing blogs: Understand your world with bing. https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/ (2013). Accessed 2019-05-27
120. Raviv, H., Carmel, D., Kurland, O.: A ranking framework for entity oriented search using Markov random fields. In: Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search (JIWES 2012), pp. 1–6 (2012). DOI 10.1145/2379307.2379308
121. Raviv, H., Kurland, O., Carmel, D.: The cluster hypothesis for entity oriented search. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information etrieval (SIGIR 2013), p. 841 (2013). DOI 10.1145/2484028.2484128
122. Reinanda, R., Meij, E., Pantony, J., Dorando, J.J.: Related entity finding on highly-heterogeneous knowledge graphs. In: IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018, Barcelona, Spain, August 28-31, 2018, pp. 330–334 (2018). DOI 10.1109/ASONAM.2018.8508650
123. van Rest, F.: A mathematical approach to scalable personalized PageRank. Bachelor thesis, Mathematisch Instituut, Universiteit Leiden (2009). URL https://www.math.leidenuniv.nl/scripties/vanRestBach.pdf
124. Richardson, M., Domingos, P.M.: Markov logic networks. Machine Learning **62**(1-2), 107–136 (2006). DOI 10.1007/s10994-006-5833-1
125. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval **3**(4), 333–389 (2009). DOI 10.1561/1500000019
126. Rousseau, F., Vazirgiannis, M.: Graph-of-word and TW-IDF: new approach to ad hoc IR. In: Q. He, A. Iyengar, W. Nejdl, J. Pei, R. Rastogi (eds.) 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013, pp. 59–68. ACM (2013). DOI 10.1145/2505515.2505671
127. Saerens, M., Fouss, F.: HITS is principal components analysis. In: 2005 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2005), 19-22 September 2005, Compiegne, France, pp. 782–785 (2005). DOI 10.1109/WI.2005.71
128. Sang, E.F.T.K., Meulder, F.D.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003, pp. 142–147 (2003). URL http://aclweb.org/anthology/W/W03/W03-0419.pdf
129. Sarma, A.D., Nanongkai, D., Pandurangan, G., Tetali, P.: Distributed random walks. J. ACM **60**(1), 2:1–2:31 (2013). DOI 10.1145/2432622.2432624
130. Schuhmacher, M., Dietz, L., Ponzetto, S.P.: Ranking entities for web queries through text and knowledge. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015, pp. 1461–1470 (2015). DOI 10.1145/2806416.2806480
131. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: Issues, techniques, and solutions. IEEE Trans. Knowl. Data Eng. **27**(2), 443–460 (2015). DOI 10.1109/TKDE.2014.2327028
132. Singhal, A.: Official google blog: Introducing the knowledge graph: things, not strings. https://googleblog.blogspot.pt/2012/05/introducing-knowledge-graph-things-not.html (2012). Accessed on 2017-04-11
133. Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.P., Wang, K.: An overview of microsoft academic service (MAS) and applications. In: A. Gangemi, S. Leonardi, A. Panconesi (eds.) Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume, pp. 243–246. ACM (2015). DOI 10.1145/2740908.2742839
134. Sordoni, A., Nie, J., Bengio, Y.: Modeling term dependencies with quantum language models for IR. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013, pp. 653–662 (2013). DOI 10.1145/2484028.2484098

135. Sowa, J.F.: Conceptual structures: Information processing in mind and machine. Addison-Wesley (1984)
136. Tan, S., Bu, J., Chen, C., Xu, B., Wang, C., He, X.: Using rich social media information for music recommendation via hypergraph model. TOMCCAP **7**(Supplement), 22 (2011). DOI 10.1145/2037676.2037679
137. Theodoridis, A., Kotropoulos, C., Panagakis, Y.: Music recommendation using hypergraphs and group sparsity. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013, pp. 56–60 (2013). DOI 10.1109/ICASSP.2013.6637608
138. Tonon, A., Demartini, G., Cudré-Mauroux, P.: Combining inverted indices and structured search for ad-hoc object retrieval. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012, pp. 125–134 (2012). DOI 10.1145/2348283.2348304
139. Urbain, J.: User-driven relational models for entity-relation search and extraction. In: Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search, JIWES '12. Association for Computing Machinery, New York, NY, USA (2012). DOI 10.1145/2379307.2379312
140. Van, T., Beigbeder, M.: Web co-citation: Discovering relatedness between scientific papers. In: Advances in Intelligent Web Mastering, Proceedings of the 5th Atlantic Web Intelligence Conference - AWIC 2007, Fontainebleau, France, June 25 - 27, 2007, pp. 343–348 (2007). DOI 10.1007/978-3-540-72575-6_55
141. Wang, X., Tao, T., Sun, J., Shakery, A., Zhai, C.: DirichletRank: Solving the zero-one gap problem of PageRank. ACM Trans. Inf. Syst. **26**(2), 10:1–10:29 (2008). DOI 10.1145/1344411.1344416
142. Xiong, C., Liu, Z., Callan, J., Hovy, E.H.: Jointsem: Combining query entity linking and entity based document ranking. In: E. Lim, M. Winslett, M. Sanderson, A.W. Fu, J. Sun, J.S. Culpepper, E. Lo, J.C. Ho, D. Donato, R. Agrawal, Y. Zheng, C. Castillo, A. Sun, V.S. Tseng, C. Li (eds.) Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017, pp. 2391–2394. ACM (2017). DOI 10.1145/3132847.3133048
143. Xiong, S., Ji, D.: Query-focused multi-document summarization using hypergraph-based ranking. Inf. Process. Manage. **52**(4), 670–681 (2016). DOI 10.1016/j.ipm.2015.12.012
144. Yang, R., Xiao, X., Wei, Z., Bhowmick, S.S., Zhao, J., Li, R.: Efficient estimation of heat kernel pagerank for local clustering. In: Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019, pp. 1339–1356 (2019). DOI 10.1145/3299869.3319886
145. Yeh, E., Ramage, D., Manning, C.D., Agirre, E., Soroa, A.: Wikiwalk: Random walks on wikipedia for semantic relatedness. In: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, August 7, 2009, Singapore, pp. 41–49. The Association for Computer Linguistics (2009). URL `https://www.aclweb.org/anthology/W09-3206/`
146. Yi, M.: Information organization and retrieval using a topic maps-based ontology: Results of a task-based evaluation. JASIST **59**(12), 1898–1911 (2008). DOI 10.1002/asi.20899
147. Yilmaz, E., Aslam, J.A.: Estimating average precision with incomplete and imperfect judgments. In: Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006, pp. 102–111 (2006). DOI 10.1145/1183614.1183633
148. Zhang, Z., Wang, L., Xie, X., Pan, H.: A graph based document retrieval method. In: 22nd IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD 2018, Nanjing, China, May 9-11, 2018, pp. 426–432 (2018). DOI 10.1109/CSCWD.2018.8465295
149. Zhiltsov, N., Agichtein, E.: Improving entity search over linked data by modeling latent semantics. In: 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013, pp. 1253–1256 (2013). DOI 10.1145/2505515.2507868
150. Zhong, J., Zhu, H., Li, J., Yu, Y.: Conceptual graph matching for semantic search. In: U. Priss, D. Corbett, G. Angelova (eds.) Conceptual Structures: Integration and Interfaces, 10th International Conference on Conceptual Structures, ICCS 2002, Borovets, Bulgaria, July 15-19, 2002, Proceedings, *Lecture Notes in Computer Science*, vol. 2393, pp. 92–196. Springer (2002). DOI 10.1007/3-540-45483-7_8

151. Zhong, M., Liu, M.: Ranking the answer trees of graph search by both structure and content. In: Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search, pp. 1–3. Association for Computing Machinery, New York, NY, USA, Portland, OR, USA (2012). DOI 10.1145/2379307.2379314
152. Zhou, M.: Entity-centric search: Querying by entities and for entities. Ph.D. thesis, University of Illinois at Urbana-Champaign (2014). URL http://hdl.handle.net/2142/72748
153. Zhu, H., Zhong, J., Li, J., Yu, Y.: An approach for semantic search by matching RDF graphs. In: S.M. Haller, G. Simmons (eds.) Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, May 14-16, 2002, Pensacola Beach, Florida, USA, pp. 450–454. AAAI Press (2002). URL http://www.aaai.org/Library/FLAIRS/2002/flairs02-088.php
154. Zhu, Y., Yan, E., Song, I.: A natural language interface to a graph-based bibliographic information retrieval system. Data Knowl. Eng. **111**, 73–89 (2017). DOI 10.1016/j.datak.2017.06.006
155. Zou, X.: A survey on application of knowledge graph. In: Journal of Physics: Conference Series, vol. 1487, p. 012016. IOP Publishing (2020). DOI 10.1088/1742-6596/1487/1/012016. URL https://doi.org/10.1088/1742-6596/1487/1/012016

A Overview of entity-oriented search approaches and tasks

Table 2: Classical information retrieval models applied to entity-oriented search.

Approach	Task(s)	Description
Virtual documents	Ad hoc entity retrieval	**Bautin and Skiena [21].** Time-dependent concordances for entities (i.e., concatenation of sentences containing the entity, optionally for a given period of time). **Dietz et al. [46].** Knowledge portfolios used to establish query-specific collections of entities and text passages relevant to the queries. Retrieval based on textual passages or Wikipedia pages for the entities in the query.
	Ad hoc entity retrieval; Attribute retrieval; Relation retrieval	**Pound at al. [118].** Defined five query categories: entity, type, attribute, relation and keyword. Index RDF using an inverted index, computing IDF per RDF property, as opposed to the whole collection. Used TF-IDF for ranking.
Combined data	Ad hoc document retrieval; Ad hoc entity retrieval	**Bhagdev et al. [26].** Documents identified by a URI and indexed using an inverted index. Entities stored in a triplestore with provenance linking to document URIs. Their hybrid search approach consisted of either document retrieval informed by entities, entity retrieval informed by documents, or a combination of both. **Bast and Buchhold [18].** Joint index for ontologies and text, based on context lists and ontology relation lists. Context lists map words or entities to text postings, by their prefixes, while ontology relation lists map source entities to target entities, along with an optional relation score.
	Ad hoc document retrieval; Ad hoc entity retrieval; Related entity finding	**Zhou [152].** Querying by entities: entities as input and documents or entities as output; entities represented by their Wikipedia pages. Querying for entities: keywords or entities as input and entities as output; proposed the CQL language over a joint index and a contextual index. Querying by entities and for entities: entities as input and output; proposed a framework analogous to related entity finding [12][§4.4.3].
	Entity list completion	**Bron et al. [31].** In order to retrieve related entities, they proposed three approaches: text-based (using the given textual description as input), structure-based (using the the given example entities as input) and a combination of both, which outperformed one isolated.
	–	**See also:** Tonon et al. [138], Xion et al. [142].
Probabilistic graphical models	Ad hoc entity retrieval	**Koumenides and Shadbolt [88].** Bayesian network to establish dependencies between entities and property instances, between property instances and property identifiers and, finally, between terms in the literal space and property identifiers. Entity search carried through Bayesian inference **Raviv et al. [120].** Markov network to model the undirected dependencies between the query and the entity. Captured the dependencies between a virtual document (representing the entity) and the query, between the entity type and the query target type, and between the entity name and the query.
	Sentence retrieval	**Urbain [139].** Markov network to model the undirected dependencies between the query and the sentence. Several models were tested, with different feature functions: aggregate (entity, sentence terms, document terms), term (sentence term, document term), entity, entity-relation and relation.
Cluster hypothesis	Ad hoc entity retrieval	**Raviv et al. [121].** Verified the cluster hypothesis for entity-oriented search: closely related entities have a high probability of also being relevant to the query. This is important for instance when implementing graph-based approaches that rely on distance.

Table 3: Learning to rank models for entity-oriented search.

Approach	Task(s)	Description
Semantic-driven	Sentence retrieval	**Chen et al. [38].** Explored explicit semantic analysis (ESA) and word2vec skip-gram as query-sentence similarity features. Used Metzler-Kanungo features as the baseline and experimented with linear regression, coordinate ascent and MART. Combining all features yielded the best results, with ESA distinguishing itself positively.
	Related entity finding	**Lin et al. [92].** Retrieved the source entity homepages based on a given narrative illustrating the relation to a target entity of a given type. Applied entity extraction, obtaining candidate target entities and computed several source-target entity-entity features, like frequency, proximity and semantic similarity. Experimented with three SVMs: (*i*) using default hyperparameters, (*ii*) using tuned hyperparameters, and (*iii*) using feature selection.
	Ad hoc entity retrieval	**Schuhmacher et al. [130].** Given a keyword query, ad hoc entity retrieval was implemented through: (*i*) document ranking; (*ii*) entity linking; and (*iii*) entity ranking. Features included mention frequency, as well as query-mention, query-entity and entity-entity similarities. A semantic kernel was used for the latter. Learning to rank models slightly improved individual feature baselines.
Virtual documents	Ad hoc entity retrieval	**Chen et al. [37].** Compared a fielded sequential dependence model (FSDM; baseline) with pairwise (RankSVM) and listwise (coordinate ascent) methods. Features included a language model, BM25, coordinate match, cosine similarity, SDM and FSDM. Results were consistently better for learning to rank over several test collections. They also found related entity names to be a fundamental field, except for question answering, highlighting the importance of training several models per query type.
Representation learning	Ad hoc entity retrieval	**Gysel et al. [73].** Tackled the keyword-based entity retrieval problem by learning a common embedding for words and entities, called latent semantic entity (LSE). They then used learning to rank based on the embeddings, but the embeddings could just as easily be applied to the vector space model. Their best feature configuration included LSE, along two other features.
–	–	**See also:** Reinanda et al. [122]

Table 4: Graph-based models for entity-oriented search.

Approach	Task(s)	Description
Link analysis*	Node importance	**Kleinberg [86].** Hypertext Induced Topic Selection (HITS) provides two scores for a node: hub and authority. The hub score is higher when a node links to multiple nodes or to highly authoritative nodes. The authority score is higher when a node receives multiple links or those links are from well-renowned hubs. HITS is usually computed for a query-dependent graph. **Page and Brin [116,30].** PageRank measures the importance of a node based on the importance (and number) of incoming nodes. PageRank is usually computed for a query-independent graph (e.g., web graph).
	Node relatedness	**Van and Beigbeder [140].** Explored bibliographic coupling (shared outgoing links) and co-citation (shared incoming links) as reranking strategies. In practice, two nodes were related based on the similarity of their immediate neighborhood.
	Node importance; Node relatedness	**Ito et al. [82].** Explored von Neumann kernels as a unified framework for measuring importance and relatedness. Also proposed Laplacian kernels and heat kernels as a way to control the bias between relatedness and importance, and to tackled limitations of bibliographic coupling and co-citation approaches.
	Node importance; Graph partitioning	**Chung [40].** Proposed the heat kernel PageRank, building on PageRank's alternative notation [123][§1.5]. Local applications can be used to identify node clusters, while global applications to measure node importance. **Kloster and Gleich [87].** Explored the heat kernel PageRank as a community detection algorithm, solving the exponential of the Markov matrix using a Taylor polynomial approximation. **Yang et al. [144].** Proposed a more efficient approach to computing heat kernel PageRank, based on Monte Carlo random walks and a reduction of the required number of random walks.
	–	**See also:** Kandola et al. [85].
Text as a graph**	Ad hoc document retrieval	**Blanco and Lioma [27].** Document as an undirected graph of co-occurring words within a sliding window, or with an added direction based on Jespersen's rank theory of POS tags. They experimented with PageRank and indegree over the two graphs as a TF replacement, combining the score with global graph-based features (e.g., average degree). Performance was improved over TF-IDF and BM25. **Rousseau and Vazirgiannis [126].** Similar to Blanco and Lioma [27], they defined a graph-of-word of co-occurring words, but considered direction and the following terms instead of centering the sliding window on each word. They found little impact of window size (used $N = 4$) and used almost no pivoted document length normalization ($b = 0.003$).
	Ad hoc document retrieval; Text classification	**Dourado et al. [49].** Represented documents as a bag of textual graphs, weighting unigrams and bigrams by term frequency. A graph dissimilarity function was proposed to cluster subgraphs and obtain a graph-based vocabulary. Assignment to this vocabulary resulted in a matrix (each subgraph compared to all centroids), which was then collapsed into a vector to represent the document, as a pooling of graph embeddings. The resulting embedding could then be used for text retrieval and classification.

(Continued on next page)

Table 4. Graph-based models models for entity-oriented search. (Continued from previous page)

Approach	Task(s)	Description
Knowledge graphs	Ad hoc entity retrieval; Ad hoc document retrieval	**Fernández et al. [55].** Given a natural language query, triples were retrieved and, in turn, used to retrieve and rank documents, based on a semantic index that combined information from a knowledge graph and a corpus.
	Ad hoc entity retrieval; Ad hoc document retrieval (Cont.)	**Balog et al. [13].** Used keyword queries and language models to retrieve documents and entities from news collections. They defined both a document-centric and an entity-centric view on their SaHaRa system, where entities augmented documents and vice-versa.
Knowledge graphs (cont.)	Ad hoc entity retrieval	**Blanco et al. [28].** Experimented with several ways to translate an RDF graph into a multi-fielded index: horizontal (*token*, *property*, *subject*), vertical (one field per property), and reduced-vertical (*important*, *neutral* and *unimportant* groupings of properties). Ranking was based on BM25F. **Neumayer et al. [106].** Experimented with two entity models to translate an RDF graph into a multi-fielded index: unstructured (one field for all properties) and structured (four property groups: *Name*, *Attributes*, *OutRelations*, *InRelations*). Ranking was based on language models.
	Knowledge graph construction and modeling	**Byrne [33].** Used RDF to integrate structured data from relational databases (each table was considered a class), with domain thesauri (represented using the SKOS ontology), and free text (using NER to identify 11 classes of entities, and relation extraction to identify 7 predicates). She compared retrieval based on SPARQL and SQL. **Google Knowledge Graph [132].** Announced in 2012 and partly powered by Freebase. Freebase was then bought and closed by Google. Public dumps were made available and migrated to Wikidata. **Microsoft Satori [119].** Announced in 2013 and presented in KDD 2018 as a tutorial on building knowledge graphs. Focused on evaluation by correctness, coverage, freshness and usage. **Microsoft Academic Graph, by Sinha et al. [133].** It contains 80 million indexed papers and six types of entities: *#field_of_study*, *#author*, *#institution*, *#paper*, *#venue* and *#event*. Built to support academic queries, based on feeds from publishers and event web sites.
	Topic modeling	**Allahyari [4].** Proposed a method for ontology-based topic modeling, experimenting with topics as distributions over ontology concepts, as well as topics as distributions over Wikipedia categories.
	–	**See also:** Gao et al. [63]
Text to entity graph	Ad hoc entity retrieval	**Bordino et al. [29]** Serendipitous search over an entity graph extracted from Wikipedia and Yahoo! Answers.
	Ad hoc document retrieval	**Ni et al. [107].** Measuring semantic similarity based on a concept graph representing a document. Proposed *Concept2VecSim* and *ConceptGraphSim*.
Entity graph to tensor	Ad hoc entity search	**Zhiltsov and Agichtein [149].** Represented entities as a tensor of adjacency matrices (one per predicate). Using tensor factorization, they obtained a matrix of latent entity embeddings, that they used to compute similarities to the top-3 entities, boosting those entities (consistent with the cluster hypothesis).
Graph matching	Ad hoc entity retrieval	**Zhu et al. [153] and Zhong et al. [150].** Matched a query graph with an entity graph (conceptual graph; also translatable to RDF). They computed the semantic similarity based on the similarity between the nodes and edges of two conceptual graphs. The user was required to provide a set of entry nodes as part of the query.

(Continued on next page)

Table 4. Graph-based models models for entity-oriented search. (Continued from previous page)

Approach	Task(s)	Description
		Zhu et al. [154]. Translated a natural language query into a graph query using named entity recognition along with dependency parsing to extract entities and their relations. The result was translated into a graph query language for a graph database.
	Ad hoc entity retrieval; Related entity finding; Entity list completion	**Minkov and Cohen [103].** Generalized multiple personal information management tasks over an entity graph and based on keyword queries (e.g., name disambiguation, threading, grouping e-mail aliases).
	Answer tree ranking	**Zhong et al. [151].** Combined content-based and structure-based features to score answer trees that contained query keywords.
Hypergraph-based*	Joint representation	**Garshol [64].** Describes topic maps, a hypergraph of topics, their associations and occurrences. It describes it as a common reference model, able to represent controlled vocabularies, taxonomies, thesauri, faceted classification and ontologies. **Yi [146].** Compared thesaurus based information retrieval with topic maps based information retrieval, finding topic maps to outperform thesauri.
	Ad hoc document retrieval	**Bendersky and Croft [24].** Proposed the query hypergraph to represent higher-order dependencies between concepts (subsets of query terms) and a document. Ranking is done using a log-linear combination of factors, based on the factor graph representation of the hypergraph.
Hypergraph-based (cont.)*	Ad hoc entity retrieval; Joint representation	**Dietz [45].** Proposed ENT Rank for modeling entity-neighbor-text relations as a hypergraph. She transformed the hypergraph into an entity co-occurrence multigraph which was used to determine which features, from text, entites, and their relations, to combine for learning a function for entity ranking.
	Document representation	**Haentjens Dekker and Birnbaum [74].** Text As a Graph (TAG) is a document representation based on a hypergraph. It links text, document, annotation and markup nodes. It can for instance be used to represent a poem line or quatrain as hyperedges of text, where the quatrains subsume lines.
	–	**See also:** Xiong and Ji [143], Cattuto et al. [35], Bu et al. [32], McFee and Lanckriet [99], Tan et al. [136], Theodoridis et al. [137], Bellaachia and Al-Dhelaan [23], Lee-Kwang and Lee [89], Akram and Dudek [3]
Random walk based	Ad hoc entity retrieval	**Hogan et al. [78].** ReConRank is used to rank nodes in a query-dependent graph, that jointly represents RDF resources and contexts. **Balmin et al. [11].** ObjectRank is used to rank nodes in a query-dependent labeled graph. A graph is induced by each query term and PageRank is used to compute a term-based score (i.e., personalized by a term) along with a global score (i.e., without personalization). An authority transfer schema is used to introduce edge bias. **Chakrabarti [36].** HubRank provides a more efficient alternative to ObjectRank. It is based on precomputed random walk fingerprints over a subgraph limited by a set of boundary nodes (blockers and losers).
	Node importance	**Espín-Noboa et al. [51].** HopRank models human navigation on semantic networks, by taking into consideration the bias of jumping to nodes at particular distances. Node importance is adjusted accordingly. **Nie et al. [110].** PopRank assigns node importance based on information from an entity graph (object graph) and a context graph (web graph). It was used in Libra, which is now Microsoft Academic Search.

(Continued on next page)

Table 4. Graph-based models models for entity-oriented search. (Continued from previous page)

Approach	Task(s)	Description
		Delbru et al. [44]. DING (Dataset rankING) also assigns node importance based on information from an entity graph and a context graph (dataset graph based on entity links).
	Related entity finding; Entity list completion	**Musto et al. [105].** Proposed a semantics-aware personalized PageRank for recommendation over a user-item-entity graph, built by combining a user-item profile with DBpedia triples. This is similar to the tasks of related entity finding or entity list completion, if the user is abstracted as an entity.

* General (hyper)graph-based approaches and introductory concepts that can be applied to entity-oriented search (e.g., multiple PageRank adaptations to entity-oriented search are covered in Section 5.8, and the query hypergraph, which defines concepts that can be entities, is covered in Section 5.7).

** Despite not leveraging entities, these models are relevant when defining joint graph-based representations for text and entities.