Knowledge Graphs: An Information Retrieval Perspective

Suggested Citation: Ridho Reinanda, Edgar Meij and Maarten de Rijke (2020), "Knowledge Graphs: An Information Retrieval Perspective", : Vol. xx, No. xx, pp 1–153. DOI: 10.1561/XXXXXXXXX.

Ridho Reinanda
Bloomberg
rreinanda@bloomberg.net

Edgar Meij
Bloomberg
emeij@bloomberg.net

Maarten de Rijke
University of Amsterdam & Ahold Delhaize
m.derijke@uva.nl

now
the essence of knowledge
Boston — Delft

Contents

1 Introduction **3**
1.1 Motivation . . . 3
1.2 Aims . . . 4
1.3 Methodology . . . 4
1.4 Scope . . . 5
1.5 Structure . . . 5

2 Preliminaries **7**
2.1 Key concepts . . . 7
2.2 Evaluation . . . 10

3 Background on Entity Linking and Recognition **12**
3.1 Entity linking . . . 12
3.2 Entity recognition and classification . . . 22

4 Knowledge Graphs for Information Retrieval **32**
4.1 Document retrieval . . . 34
4.2 Entity retrieval . . . 44
4.3 Entity recommendation . . . 52
4.4 Entity relationship explanation . . . 60
4.5 Conclusion . . . 64

5 Information Retrieval for Knowledge Graphs **65**
5.1 Entity discovery . . . 68
5.2 Entity typing . . . 74
5.3 Entity-centric document filtering . . . 79
5.4 Relation extraction and link prediction . . . 84
5.5 KG quality estimation . . . 95
5.6 Conclusion . . . 102

6 Applications **103**
6.1 Web search . . . 103
6.2 Knowledge graph construction . . . 110

7 Conclusion and Discussion **116**
7.1 Conclusion . . . 116
7.2 Discussion . . . 117

Acknowledgements **122**

Appendices **123**

A Acronyms used **124**

B Resources **127**
B.1 Corpora . . . 127
B.2 Knowledge graphs . . . 127
B.3 Datasets . . . 127
B.4 Code . . . 131
B.5 Libraries . . . 131
B.6 Tutorials . . . 131

References **134**

Knowledge Graphs: An Information Retrieval Perspective

Ridho Reinanda[1], Edgar Meij[2] and Maarten de Rijke[3]

[1] *Bloomberg L.P.; rreinanda@bloomberg.net*
[2] *Bloomberg L.P.; emeij@bloomberg.net*
[3] *University of Amsterdam & Ahold Delhaize; m.derijke@uva.nl*

ABSTRACT

In this survey, we provide an overview of the literature on knowledge graphs (KGs) in the context of information retrieval (IR). Modern IR systems can benefit from information available in KGs in multiple ways, independent of whether the KGs are publicly available or proprietary ones. We provide an overview of the components required when building IR systems that leverage KGs and use a task-oriented organization of the material that we discuss. As an understanding of the intersection of IR and KGs is beneficial to many researchers and practitioners, we consider prior work from two complementary angles: leveraging KGs for information retrieval and enriching KGs using IR techniques. We start by discussing how KGs can be employed to support IR tasks, including document and entity retrieval. We then proceed by describing how IR—and language technology in general—can be utilized for the construction and completion of KGs. This includes tasks such as entity recognition, typing, and relation extraction. We discuss common issues that appear across the tasks that we consider and identify future directions for addressing them. We also provide pointers to

Ridho Reinanda, Edgar Meij and Maarten de Rijke (2020), "Knowledge Graphs: An Information Retrieval Perspective", : Vol. xx, No. xx, pp 1–153. DOI: 10.1561/XXXXXXXXX.

datasets and other resources that should be useful for both newcomers and experienced researchers in the area.

1

Introduction

1.1 Motivation

A *knowledge graph* (KG) is a repository of entities as well as their relationships and attributes that is represented as a graph. In modern approaches to information access, KGs are ubiquitous (Dalton and Dietz, 2013). Specifically, in information retrieval (IR) KGs are instrumental in enabling semantic search.

There are two hallmarks of semantic search in an IR context: (1) going beyond "ten blue links" in order to return relevant results of any kind (such as direct answers, actionable entities, or relationships) and (2) understanding queries and documents, and improving the matching between them with relevant relationships. Ideally, a search engine is able to directly answer a user's information need—or at least generate possible interpretations of the information need that is expressed through the query. To achieve this goal, various entity-oriented components that solve specific problems at different stages in the information retrieval pipeline are required, including identifying entities in the query, identifying entities in documents, and methods that leverage entity and relationship information to help identify relevant items to retrieve. +

Despite the fact that IR and KGs are increasingly intertwined in

the context of modern web and domain-specific search engines, there is no broad treatment in the literature of KGs from an IR perspective, and vice versa. We aim to fill this gap through this survey by providing a task-oriented overview of research in this area.

1.2 Aims

The aim of this survey is to bridge two important components of modern information access: IR and KGs. We summarize research work, group related approaches, and discuss challenges shared across tasks at the interface of IR and KGs. Our contributions in this survey can be summarized as follows: (1) we present an extensive overview of tasks related to KGs from an IR perspective; (2) we provide a thorough review for each task; and (3) we present discussions on common issues that are shared among the tasks.

1.3 Methodology

To meet the aim articulated above and to be able to present the methods described in this survey in a systematic manner, we first identify different sets of tasks related to IR and KGs and group individual tasks that are closely related. The main organizational principle that we use in the survey is to group tasks in two directions: *knowledge graphs for information retrieval* and *information retrieval for knowledge graphs*.

For each task, we trace back its origin, the original motivation, setup, and define the task in a formal fashion. We then identify seminal work or influential approaches as they have been introduced over time. We group approaches based on characteristics that are natural for each task. We also identify related work based on these groupings. We put more emphasis on recent developments concerning the task, how the methods differ from early approaches, and the interesting additional problems that arise over time in the context of the task.

Having examined each task one-by-one, we then proceed to identify the key challenges that we encounter frequently across tasks. We focus on challenging aspects that will be beneficial for future research.

1.4 Scope

We consider over 300 publications published prior to 2020 and spanning the fields of information retrieval, knowledge representation, machine learning, and natural language processing. Due to the broad nature of the survey, we put more emphasis on recent advances involving new tasks and approaches. Thus, some tasks and approaches will be covered in greater detail than others.

In the survey, our view of IR is an inclusive one and that incorporates natural language processing and language technology techniques. We also consider tasks that have an origin in those fields, such as entity recognition, relation extraction, and knowledge base (KB) completion.

Some of the tasks that we consider cover a broad area. For broad tasks—that may well deserve a survey of their own—we only cover key publications and present the task in a high-level fashion. This includes tasks such as entity recognition, entity linking, and relation extraction. We present an overview of tasks, but refer the reader to existing surveys (if they exist) for details. For emerging, specific tasks we provide more details in addition to a literature review. We present their setup and contrast different approaches with more depth and detail.

Recent interest in the area of semantic search has not only given rise to hundreds of publications but also to attempts to synthesize the material. By now there is a growing number of tutorials in the area, which we enumerate in Appendix B.6. While we believe that ours is the first survey to focus on the interaction between IR and KGs, there is a recent survey on semantic search by Bast *et al.* (2016) that partially overlaps with ours. The most significant differences are that we discuss recent developments on how KGs are being leveraged for IR, we provide a broader coverage of knowledge graph construction and completion, and finally, we present several applications that involve a combination of the individual tasks and components in our survey.

1.5 Structure

The rest of this survey is organized as follows. In Chapter 2 we describe the background: definitions of fundamental concepts and notation that

we use throughout the survey. In Chapter 3 we introduce core entity-related tasks on which we build in the remainder of the survey: entity linking and named entity recognition and classification. The heart of the survey consists of Chapters 4 and 5. Chapter 4 describes how KGs are being leveraged to improve IR tasks. In Chapter 5 we turn the table and detail how IR and, more generally, language technology is being used for the construction and completion of KGs. Chapter 6 is meant to provide detailed motivation for the survey by offering treatments of end-to-end tasks at the interface of IR and KGs. We conclude the survey in Chapter 7 with a look back, with a discussion of the key issues that we identified in the course of the survey, and with potential research directions in at the interface of IR and KGs.

Acronyms used and useful resources used in this survey are listed in appendices to this survey, Appendix A and B, respectively.

As to possible reading orders of the material in the survey, we recommend the following. Readers who are relatively new to the area should simply read all chapters in their natural order: 1, 2, ..., 7. Readers who are already familiar with the area can move ahead to the core of the survey in Chapter 4. Alternatively, they can freshen up on notation and terminology in Chapter 2, catch up on the background material on entity linking and entity recognition and classification in Chapter 3, sample from Chapter 6, and then continue with the remaining material. See Figure 1.1.

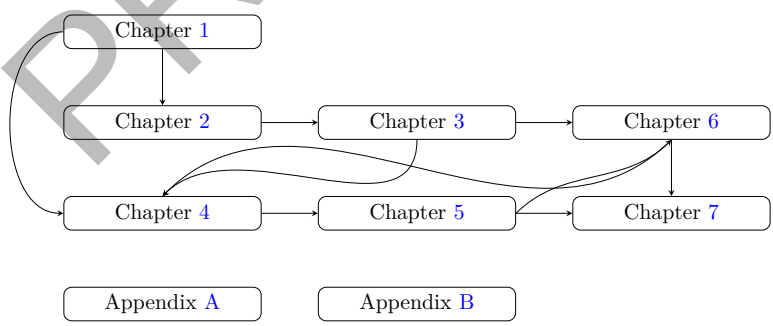

Figure 1.1: Possible reading orders.

2

Preliminaries

In this section we briefly introduce the core concepts that we use in the remainder of the survey.

2.1 Key concepts

Given that we present work at the intersection of multiple research areas, it is important to introduce a unified set of definitions and terminology. Our definitions are based mainly on (Kripke, 1980; Sekine, 2009; Meij *et al.*, 2013).

Definition 2.1.

- An *entity* e is an atomic, identifiable object that has a distinct and independent existence.
- A *named entity* is a specific entity for which one or many designators or proper names can be used to refer to it.
- An *entity type* t is a set of classes that is appropriate for an entity based on a pre-defined class hierarchy.
- A *mention* is a text segment that refers to an entity.
- A *relationship type* is a type of connection between two entities.
- A *relation* r is an instance of a relationship type between two

Table 2.1: Examples.

Definition	Example
entity	city
named entity	London
entity type	city
mention	London
relation type	capitalOf
(*relation*)	(London, capitalOf, UnitedKingdom)
attribute	city:population
knowledge base	Wikipedia
knowledge graph	Wikidata

entities, the nature of which can be defined with a label and, can be complemented with attributes and attribute values.

- An *attribute* is a specific characteristic or property of an entity or relationship, with zero or more values.
- A *knowledge base* is a repository of entities with information about their relationships and attributes in a (semi-)structured format.
- A *knowledge graph* is a knowledge base that is specifically represented as a graph. In a knowledge graph, entities, attributes, and relations are represented through the nodes and edges in the graph. Entities are typically represented as nodes, while relationships are represented as edges.
- An *entity profile* is a textual description of an entity.

Examples for each of the definitions are given in Table 2.1.

The notation used in the survey is collected in Table 2.2. Below, and especially in Chapters 4 and 5, we will introduce further terminology where needed. For now, consider the example in Figure 2.1, where we depict a fragment of a knowledge graph from the movie domain, demonstrating the types of a number of example entities (e.g., *actor* and *singer*), the relationships between them, and entity attributes, e.g., *date of birth*.

KGs can be obtained in different ways. They can be created manually from scratch, either by crowdsourcing or by experts. There are also

Table 2.2: Notation used in the paper.

Notation	Description
e	entity
f	fact
m	mention
d	document
q	query
r	relation
s	text segment
t	entity type
T	entity classification system

many publicly-available, open-domain KGs that can be used, including Wikidata and Freebase. See Appendix B.2 for a more detailed list.

KGs can also be generated from structured or semi-structured sources such as Wikipedia infoboxes (Lehmann *et al.*, 2015), tables (Dong *et al.*, 2014b), or social networks (Brambilla *et al.*, 2018). Alternatively, through information extraction they may be obtained from unstructured textual sources, such as the web (Dong *et al.*, 2014b; Dong *et al.*, 2014a; Banko *et al.*, 2007; Fader *et al.*, 2011), social media (Brambilla *et al.*, 2017), news articles (Ji *et al.*, 2011; Surdeanu *et al.*, 2011; Kuzey *et al.*, 2014), or scientific articles (Fathalla *et al.*, 2017). Chapter 5 provides more background on some of the techniques used in this context. In addition, dialogues (Li *et al.*, 2014) and multimedia content (images, videos) have also been used to construct or populate KGs (Melo and Tandon, 2016; Zhu *et al.*, 2015).

KGs are useful in different settings and in different roles. Besides the IR-centered techniques listed in Chapter 4, KGs can also support users to explore information using entities and relations as navigational aids (Sarrafzadeh *et al.*, 2014) and for decision support (Zhang *et al.*, 2017a). Moreover, KGs feed conversational search interfaces (Hakkani-Tür *et al.*, 2014). For more detailed accounts of the use of KGs, we refer the reader to Chapter 6, where we describe two end-to-end pipelines that utilize knowledge graphs and information retrieval techniques. The first focuses on web search and the second deals with building knowledge

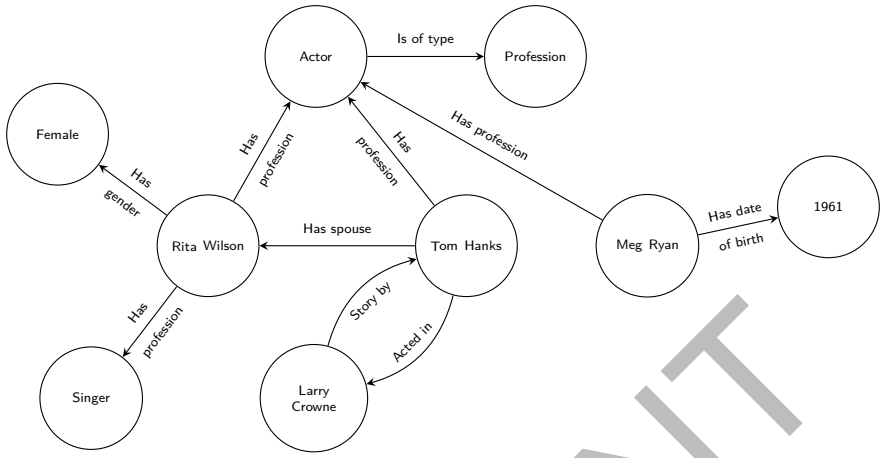

Figure 2.1: Example of a fragment of a knowledge graph in the movie domain.

graphs from scratch using unstructured text in a document corpus.

2.2 Evaluation

Several types of evaluation metrics are being used for the tasks that we will discuss later in the survey. For IR tasks we use the following standard metrics:

- Recall, used in Section 3.1;
- Precision, used in Section 3.1, 4.2, and 4.3;
- F1, used in Section 3.1;
- Mean Average Precision (MAP), used in Section 4.1, 4.2, and 4.3;
- Mean Reciprocal Rank, used in Section 4.3; and
- NDCG, used in Section 3.1, 4.1, 4.2, 4.3, and 4.4.

For details, we refer the reader to a survey on evaluation in information retrieval, such as (Sanderson, 2015), or to a standard textbook on information retrieval, such as (Manning *et al.*, 2008; Croft *et al.*, 2009). As we will point out in Chapter 4, some non-standard metrics have also been used in the context of semantic search. These include serendipity (SRDP) (Bordino *et al.*, 2013b) in Section 4.3.

As we will see in Chapter 5, many tasks related to KG construction can be understood as classification tasks, where the items to be classified are text segments, entities or documents. Hence, frequently used metrics

include:

- Recall, used in Section 3.2, 5.1, and 5.3;
- Precision, used in Section 3.2, 5.1, and 5.3; and
- F1, used in Section 3.2, 5.1, and 5.3.

A special variant of F1 was considered in the document filtering task in Section 5.3.

Over the years, many datasets have been released for tasks listed later in this survey, both for IR tasks and for tasks related to KG construction. Those tasks will be introduced in Chapters 4 and 5, respectively. The shared datasets used in studies of those tasks that we survey are listed in Appendix B.3.

3

Background on Entity Linking and Recognition

In this chapter we briefly recall core entity-related tasks on which we build in the remainder of the survey, i.e., entity linking and named entity recognition and classification.

We start our discussion with entity linking (Section 3.1). Even though this might not be considered a core IR task, we show that it is essential in any approach that uses KGs to improve the effectiveness of an IR system. We then continue to discuss entity recognition and classification in Section 3.2.

3.1 Entity linking

The goal of entity linking is to provide a form of "semantic grounding" of a text using entities in a KG, by determining which textual spans refer to which specific entities. Entity linking has its roots in the domains of natural language processing (where it evolved from cross-document coreference resolution) and databases (where it is known as record linkage) (Meij *et al.*, 2013; Dietz *et al.*, 2018). The specific task of linking mentions to Wikipedia was popularized by Mihalcea and Csomai (2007) and later developed as one of the main tasks in the Text Analytics Conference (TAC), where the focus was on evaluating and improving

Knowledge Base Population (KBP) (Ellis *et al.*, 2014). Entity linking is formally defined as follows:

Definition 3.1 (Entity linking)**.** Given a text, detect segments of entity mentions m within the text, and link them to an entity e in a knowledge graph KG.

One often-used way of detecting segments of entity mentions is using named entity recognition, which we discuss in Section 3.2.

3.1.1 Approaches to entity linking

Approaches to entity linking can be categorized into *feature-based, graph-based, neural, and joint* approaches. Before we dive into methods for performing entity linking, we first need to define how success is measured and how entity linking systems are evaluated. An entity linking system typically consists of two stages: *mention detection* and *disambiguation*, i.e., identifying the substrings that may denote entities and subsequently linking each mention to a specific entity (Ratinov *et al.*, 2011). The evaluation of an entity linking system therefore happens at either the document level ("List all entities that are mentioned in this input text") or at the mention level ("List all mentions in this input text and identify the most likely entity for each"). Note that in some variants of the latter case the mentions are given and that it may also be possible to identify so-called "Nil" entities, i.e., entities that do not exist in the KG (yet).

Given a relative lack of standardized test collections and, hence, an abundance of non-comparable results, some researchers have started to standardize entity linking approaches and experimental frameworks. Cornolti *et al.* (2013) design and implement a benchmarking framework for fair and exhaustive comparisons of entity linking systems. The framework is based on the definition of problems related to the entity linking task, a set of measures to evaluate system performance, and a comparative evaluation involving all publicly available datasets, containing texts of various types such as news, tweets, and web pages. They conduct extensive comparisons of all entity linkers available at the time. Hachey *et al.* (2014) propose a shared evaluation paradigm for the task of entity recognition and disambiguation. They review and compare

evaluation regimes found in the literature. The evaluation software and standardized system outputs are provided online. Usbeck *et al.* (2015) have taken these meta-evaluations and developed an online framework called Gerbil to benchmark entity linking approaches.

Feature-based approaches to entity linking

Early work on entity linking introduces features that have become commonplace in both mention detection and disambiguation. Mihalcea and Csomai (2007) introduce the notion of *keyphraseness*: the probability of a phrase to be selected as a keyword, i.e., a mention, in a document. Medelyan *et al.* (2008) propose the notion of *commonness*: the relative popularity of each candidate entity as a target given the same mention. When combined, these two features already constitute a simple but working end-to-end entity linking system.

To evaluate their approach, Mihalcea and Csomai (2007) construct a gold standard test collection using Wikipedia articles. From a set of keywords manually selected by Wikipedia contributors, they evaluate the performance of their disambiguation method for linking these keywords. Their main finding is that a KG-based method, i.e., based on a notion of context similarity of Wikipedia articles, is orthogonal to that of a feature-based method with context words and sense features.

Milne and Witten (2008) introduce another machine learning-based method based on the notion of *relatedness*: the semantic similarity between two entities (as defined by using Wikipedia articles). The semantic similarity is computed as a function of their incoming and outgoing intra-Wikipedia links. Milne and Witten (2008) were the first to propose a machine learning approach to entity linking by combining *commonness* and this *relatedness* metric. For evaluation, they use a test collection based on news articles from the AQUAINT text corpus. For training, they sample a number of Wikipedia articles of the same length as the news articles and use the links created by Wikipedia editors. The approach introduced by Milne and Witten (2008) outperforms the work by Medelyan *et al.* (2008) and Mihalcea and Csomai (2007). One important takeaway from the experiments of Milne and Witten (2008) is that commonness is a strong baseline feature for this task.

Maximizing the relatedness of relevant entities will minimize disambiguation errors. Based on this notion, Ceccarelli *et al.* (2013) address the problem of learning entity relatedness functions to improve entity linking. They formalize the problem of learning entity relatedness as learning a ranking function and show that their machine-learned function performs better than previously proposed relatedness functions. Furthermore, they show that improving this ranking-based relatedness function also improves the performance of state-of-the-art entity linking algorithms. Similar to (Ceccarelli *et al.*, 2013), Charton *et al.* (2014) leverage mutual disambiguation for entity linking, based on the idea that entity linking should maximize the relatedness of the entities in the candidate set. The supervised approach introduced by Ceccarelli *et al.* (2013) improves the linking performance of graph-based methods (Ferragina and Scaiella, 2010) and feature-based methods (Milne and Witten, 2008). The improvements are achieved by replacing the original relatedness function in both methods with a learned relatedness function. The overall improvements obtained are in the range of 1–10% in terms of normalized discounted cumulative gain (NDCG).

Meij *et al.* (2012) present a machine learning-based method for entity linking on tweets, incorporating commonness and other features. Similar work is presented in (Guo *et al.*, 2013). When performing entity linking in microblog posts, they leverage additional resources, in particular, extra posts. First, they expand the input post context with similar posts, i.e., they construct a query with the given post and search for similar posts. Disambiguation benefits from the extra posts if these posts are related to the given post in context, providing additional signals for disambiguation. Ran and Wang (2018), on the other hand, specifically address the limited extendability and scalability for entity linking on tweets. They propose a disambiguation method based on factor graphs and achieve linear complexity with respect to the number of mentions during the disambiguation step.

Graph-based approaches to entity linking

Some approaches perform entity disambiguation collectively, optimizing the coherence between candidate entities. The intuition is that a related

set of entities will provide context for disambiguation, as they tend to appear together in the text. One way to optimize for such coherence is by employing a graph-based approach.

The following are seminal graph-based publications in entity linking. Hoffart *et al.* (2011) combine three important intuitions: the prior probability of an entity, the similarity between the context of a mention and a candidate entity, and also the coherence among candidates entities for all mentions together. They build a weighted graph of mentions and candidate entities, maximizing coherence by computing a dense subgraph that approximates the best joint mention-entity mapping. Other seminal publications include (Ferragina and Scaiella, 2010) and (Ratinov *et al.*, 2011). Ferragina and Scaiella (2010) focus on short text fragments, opening up a line of work around entity linking on tweets (Meij *et al.*, 2012), snippets of search results (Cornolti *et al.*, 2016), and news feed items (Fetahu *et al.*, 2015). Their method, *TagMe*, addresses the ambiguity of mention-to-entity mappings by finding a collective agreement among them and maximizing their coherence. Ratinov *et al.*, 2011 propose an approach that casts the entity linking task as finding a many-to-one matching on a bipartite graph of entities and mentionds. In addition to coherence in a global context, their approach also takes into account local features, similar to the *feature-based approaches* to entity linking mentioned above.

Alhelbawy and Gaizauskas (2014) address the disambiguation problem collectively by representing candidate entities as nodes and associations between different candidates as edges between the nodes. They rank the nodes with PageRank and combine it with an initial confidence score for candidate selection. Also, Ganea *et al.* (2015) introduce a probabilistic entity linking approach that disambiguates entity mentions collectively. Disambiguation is performed by considering both the prior of entity occurrences and local information extracted from words surrounding the mentions. They rely on loopy belief propagation to perform approximate inference. Their approach relies on three sources of information: a probabilistic name-to-entity mapping derived from a large corpus of hyperlinks, pairwise co-occurrence estimated from a large corpus, and contextual entity words statistics.

A context around the mentions that is processed during entity

linking may contain noisy, uninformative words. To address this, Lazic *et al.* (2015) propose a probabilistic model for entity linking that is designed to be resilient to non-relevant context features. In addition, they supplement labeled training data with a large unlabeled text corpus. The unlabeled data is used to re-estimate the parameters of their context model.

On a related note, Globerson *et al.* (2016) explore an attention-based mechanism for improving coherence. The main intutition is that coherence should not be considered for all pairs of entities in a document, but rather focused on a small number of strong relations involving salient entities in the document. Building on the mention and context model proposed in Lazic *et al.* (2015), they add a soft attention component to capture this notion.

The method introduced by Ganea *et al.* (2015) is one of the current state-of-art graph-based models. Their method outperforms other graph-based entity linking methods, including AIDA (Hoffart *et al.*, 2011), TagMe (Ferragina and Scaiella, 2010), and non-graph based approaches, like Wikipedia Miner (Milne and Witten, 2008). State-of-the-art performance is achieved by exploiting entity co-occurrence statistics in a fully probabilistic manner. The graph-based method in (Alhelbawy and Gaizauskas, 2014) performs slightly better than (Hoffart *et al.*, 2011), but the performance was reported in terms of accuracy, so it is not comparable to the performance of (Ganea *et al.*, 2015).

Neural approaches to entity linking

With the proliferation of deep learning applications, several entity linking methods that use neural architectures have been proposed. Cai *et al.* (2015) propose an entity disambiguation model based on deep neural networks. Instead of utilizing simple similarity measures and their combinations, they directly optimize the document and entity representations. Their approach utilizes auto-encoders to learn an initial document representation in an unsupervised manner (pre-training). This is later followed by a supervised training step to improve the representation based on a given similarity measure, to make sure that similar entities in this measure has similar representations.

As with other neural approaches, the obvious advantage lies in the absence of feature engineering. Cai *et al.* (2015) compare their approach with collective disambiguation methods; it outperforms complex collective disambiguation approaches such as those presented in (Shirakawa *et al.*, 2013) and (Kulkarni *et al.*, 2009). One key takeaway from their work is that collective evidence for disambiguation can only help when local evidence (consisting of context words) is not sufficient. Additionally, the best performance is obtained when the similarity score obtained by their approach is combined with a collective framework.

Liao *et al.* (2017) use deep neural nets to map queries and candidate reference entities to feature vectors in a latent semantic space where the distance between a query and its correct reference entity is minimized. They also utilize web search result information to help generate large amounts of weakly supervised training data (similar to (Cornolti *et al.*, 2016)) for their training process. Unfortunately, they do not use any standard benchmarks for evaluation. Similarly, Zhu and Iglesias (2018) leverage the semantic similarity between a mention's context and each candidate entity's type that is measured using a variant of word2vec. Finally, Gupta *et al.* (2017) present a neural entity linking system that learns a unified dense representation for each entity using multiple sources of information including its description, contexts around its mentions, and its types. They evaluate their method on a number of common test collections and find improvements over several baselines including those in (Hoffart *et al.*, 2011).

Kolitsas *et al.* (2018) introduce a neural end-to-end approach to entity linking which performs both mention detection and named entity disambiguation in a joint fashion. The approach considers word-character embeddings, mention embeddings, and also entity embeddings. The initial input, word-character embeddings are conatenated from word embedding and character embeddings which can capture important word lexical information. Next, context-aware embeddings for each word are learned over the word-character embeddings using a bi-LSTM layer. Then, each mention embeddings vector is constructed as a fixed size representations obtained from the concatenation of the context-aware word embeddings of the first, last, and "soft-head" words in the mention ("soft head" is a task specific head word learned using attention mech-

anism). The entity embeddings is pre-trained based on entity-word co-occurrence, similar to *word2vec* (Mikolov *et al.*, 2013).

For each possible entity-mention pair, a final local score is then computed based on the similarity between mention representation and entity embeddings combined with prior linking probability and also long range attention scores, combined through a feed-forward neural network. finally, The model also incorporates a global score, to ensure the coherence between candidate linking. During training, all spans of text are considered as negative examples, with a known gold mentions as negative examples. Even though Kolitsas *et al.* (2018) do not improve over all earlier methods when compared on various test collections, their proposed method that leverages word, entity, and mention embeddings does exhibit strong performance overall. Furthermore, the end-to-end nature of their solution is attractive

Joint approaches to entity linking

The final category of approaches to entity linking that we discuss here aims to jointly perform entity recognition and linking. Cross-document coreference resolution is a task that is closely related to entity-linking. The goal in this task is to compute equivalence classes of mentions that denote the same entity in a document corpus, without explicitly linking them to a knowledge graph entry as is done with entity linking.

Dutta and Weikum (2015) jointly solve the problem of cross-document coreference resolution and entity linking. Their method is *unsupervised*, where the output of coreference resolution informs entity linking and vice versa. The coreference resolution and linking steps are performed alternately in an iterative fashion that focuses on the highest-confidence unresolved mentions. Sil and Yates (2013) propose a re-ranking approach for joint entity recognition and linking; they retrieve a large set of candidate mentions from a entity recognition system and candidate links from an entity linking system, and then rank candidate-entity mention pairs. The joint model is used to re-rank candidate mentions and entity links produced by base recognition and linking models. Luo *et al.* (2015a) also propose a method that takes into account the mutual dependency between entity recognition and entity linking. If their entity

recognition component is highly-confident about its output of entity boundaries and types, it will encourage the linking of an entity that is consistent with this output. In Section 5.1 we discuss an approach by Mohapatra *et al.* (2013) that jointly addresses linking and discovery; it also belongs to this category.

There are several reasons why a joint approach would help improve the performance of an entity linking system. For Dutta and Weikum (2015) the global context provided by cross-document coreference resolution improves both the feature space and the performance of the entity linking component. Similarly, (Sil and Yates, 2013)'s re-ranking strategy allows for the introduction of features that represent the dependency between disambiguation and boundary detection decisions. Linking performance improves. With this joint strategy, the authors outperform popular non-joint approaches to entity linking by Milne and Witten (2008) and Ratinov *et al.* (2011). Finally, Luo *et al.* (2015a) also outperform the methods in (Hoffart *et al.*, 2011; Kulkarni *et al.*, 2009) as their approach to entity linking is able to learn the mutual dependency between the type of a recognized entity and its Wikipedia type. Note that, the neural approach introduced in (Kolitsas *et al.*, 2018) also belongs to the category of of joint approaches.

3.1.2 Relations between entity linking and other tasks

Entity linking approaches may employ an *entity recognition* system (Section 3.2) as a way of performing mention detection, and some methods perform recognition and linking jointly, as we have discussed earlier. Entity linking techniques are also important in enabling other tasks. Entity linking can help improve *document retrieval* (see Section 4.1). In principle, other tasks that rely on entity-document features can be improved by having a reliable entity linking system, as it would help reduce the noise generated by incorrect entity-document associations. Entity linking is also important for resolving entities for *relation extraction* to complete a knowledge graph (Section 5.4). As we will discuss later, linking confidence is sometimes used as a signal for entity discovery (Section 5.1)

3.1.3 Outlook on entity linking

Approaches to entity linking have evolved over time, moving from individual notions of mention-entity relevance as well as using feature-based machine learning approaches, towards applying representation learning and neural network-based models. In fact, more recent work approaches entity linking end-to-end in a deep learning framework, which allows propagating information between the mention detection and entity disambiguation subtasks. The experimental results show that such neural methods are able to outperform earlier approaches on most (but not all) test collections (Kolitsas *et al.*, 2018). Furthermore, they are able to easily incorporate additional, external information in the form of, e.g., web search results or graph-based information.

Earlier work on entity linking made clear that the networks formed not only by relationships between entities, i.e., the KG, but also by relationships between mentions in a document carry signal for disambiguation. Such graph-based methods not only help define more sophisticated notions of coherence but they may also help to focus the sets of candidate entities for mentions in a document. Most recent work explicitly encodes this kind of information in the form of graph embeddings, and shows even further improvements over earlier neural methods (Sevgili *et al.*, 2019; Gerritse *et al.*, 2020).

Interesting future directions for entity linking include improving *linking for queries* and *linking with sparse knowledge graphs*. Entity linking for queries is important as it allows for better query understanding, which in turn will help search engines to retrieve relevant information. Entity linking in queries is a challenging problem because, unlike the general setup, queries are short, written in a telegraph-style, and typically only very limited context is available (Joshi *et al.*, 2014). Meij *et al.* (2009; 2011) use a feature-based approach in conjunction with supervised machine learning, augmenting term-based features with search history-based and concept-specific features and linking entities occurring in queries to DBpedia. Pantel and Fuxman (2011) estimate the relevance of the query string of an entity from query-click graphs. Cornolti *et al.* (2016) introduce a system for linking entities mentioned in web search queries. An improved approach to linking entities in

queries using contextual information and semantic matching is presented by Blanco *et al.* (2015), who learn entity embeddings using information from Wikipedia. Their approach can naturally be extended by incorporating similar information from news, related queries, and trends—effectively leveraging such contextual signals is an important direction for future work.

The coverage of KGs may be limited in certain domains, causing issues with long tail entities. For instance, entity relatedness is an important signal for entity linking that is very sparse for such domains and entities. Developing alternative ways to infer such information from various sources and integrating those methods for linking purposes is therefore an important challenge.

3.2 Entity recognition and classification

Recognizing entities in text is a well-known problem and one of the most fundamental entity-oriented tasks. The MUC-6 task (Grishman and Sundheim, 1996) introduced the named entity extraction task and, later on, the Computational Natural Language Learning (CoNLL) (Tjong Kim Sang and De Meulder, 2003) and ACE evaluation campaigns (Doddington *et al.*, 2004) further drove research in this area. The task of named entity recognition is formally defined as follows:

Definition 3.2. Given a text segment s, the *entity recognition* task is to detect segments of entity mentions m within s. Given a type classification system T and an entity mention m within a text segment s, the *entity mention classification* task is to decide whether m belongs to a type $t \in T$ and, if so, which type.

Initially, named entity recognition focused on classifying entities into fairly generic entity types such as *person*, *organization*, *location*, and so on. Later, more fine-grained class hierarchies were proposed, for instance by Sekine and Nobata (2004), who consider 150 "extended" entity types.

3.2.1 Approaches to entity recognition and mention classification

Below, we discuss approaches to entity recognition and classification. we first discuss approaches that solve both the recognition and classification steps, and then we continue with approaches that focus on the classification step only.

The following approaches attempt to solve entity recognition and classification jointly. In their seminal paper on entity recognition and classification, Finkel *et al.* (2005) introduce the Stanford named entity recognition (NER) system that works by augmenting a conditional random fields-based entity recognition system with long distance dependency models, to account for long distance dependencies that are commonly found in natural language.

Rule-based approaches to entity recognition and classification

Early approaches to named entity recognition and classification rely on dictionaries and handcrafted rules. A typical entity recognition rule would utilize signals such as the appearance of certain phrases, word classes, part-of-speech information, and named entity tagging labels. A complex named entity tagger can be built by formulating and combining sets of these rules (Sekine and Nobata, 2004). Such methods often achieve high accuracy combined with low coverage. Furthermore, they are also very costly to create and maintain. Researchers turned to *supervised* and *semi-supervised* approaches in order to address these issues.

Feature-based approaches to entity recognition and classification

Rather than specifying complex rules manually, supervised learning approaches aim to learn to classify entities from data using contextual clues around the entity mentions. Supervised learning approaches to entity recognition utilize different classes of learning algorithms such as Hidden Markov Models (Bikel *et al.*, 1999), Decision Trees (Sekine, 1998), Maximum Entropy (Borthwick, 1999), Support Vector Machines (Asahara and Matsumoto, 2003), and Conditional Random Fields (McCallum and Li, 2003; Finkel *et al.*, 2005). The problem is

often formulated as a sequential classification problem: tagging words in a sentence sequentially to indicate whether they are a part of a named entity or not.

Feature-based approaches can also be trained in a semi-supervised fashion, starting by selecting a small number of seed entities, building contextual clues relevant to these seeds, and then generalizing the obtained patterns to recognize new entities. Approaches belonging to this category are presented in (Collins and Singer, 1999; Cucchiarelli and Velardi, 2001; Riloff and Jones, 1999; Pasca *et al.*, 2006).

Dalton *et al.* (2011) propose a context expansion method for named entity recognition based on passage retrieval. The proposed method can be incorporated into structured classification models for NER. This retrieval-based feature expansion outperforms previous aggregation models on the CoNLL 2003 test set. The authors also show that external unlabeled data can be incorporated in addition to labeled data, and that it helps to improve performance.

Embedding-based approaches to entity recognition and classification

Ren *et al.* (2015) introduce a joint approach to entity recognition and classification based on distant supervision. They perform phrase mining to generate entity mention candidates and relation phrases and enforce the principle that relation phrases should be softly clustered when propagating type information between their argument entities. The type of each mention is predicted based on type signatures of its co-occurring relation phrases and type indicators of its surface name. Ren *et al.* formulate a joint optimization problem for the type propagation and relation phrase clustering tasks. Their approach outperforms Stanford's NER system (Finkel *et al.*, 2005), on both the New York Times and Yelp corpora, achieving F_1 scores of 0.94 and 0.79 on the recognition and classification tasks, respectively. On a corpus built on tweets, their approach achieves lower precision than Stanford NER, with higher recall.

Lample *et al.* (2016) introduce two neural architectures for named entity recognition: (1) bidirectional LSTMs and conditional random fields; and (2) a transition-based approach utilizing stack LSTMs. These

models use two sources of information about words: character-based word representation learned from a supervised corpus, and unsupervised word representation learned from unannotated corpora.

Approaches that strictly focus on entity (type) classification are either *feature-based*, *embedding-based*, or *extractor-based*. All of these are further detailed below. Most of the approaches that we mention perform fine-grained classification of types.

Feature-based approaches to entity classification

Ling and Weld (2012) introduce a feature-based approach for fine-grained entity recognition based on multi-label classification. After performing entity recognition with a conditional random fields model, they employ features such as tokens, word shape, parts-of-speech tags, unigrams, or bigrams with multi-label classifiers based on perceptrons. All non-zero prediction scores are considered as relevant types for each entity mention m within a context s. The classifiers are trained with automatically generated training data. To generate this data, they utilize linked segments m_e in a sentence contained in the corresponding Wikipedia page for entity e, and retrieve the types of e from Freebase. To reduce noise, only well-maintained Freebase types with more than five instances are kept.

Ling and Weld also introduce a dataset for entity classification constructed from Wikipedia sentences. When evaluated on this dataset, their approach outperforms the Stanford NER system by ∼11%. In addition, it has been shown that incorporating type information can help to improve the performance of relation extraction systems (Section 5.4).

To address issues with out-of-context labeling of distant supervision data, Gillick *et al.* (2014) introduce context-dependent tagging. They do so by applying label pruning heuristics: removing sibling types, removing types that do not agree with coarse-grained classification, and types that do no meet some mininum occurrences in the documents. During training, Gillick *et al.* leverage features such as head/non-head words, cluster id, character trigrams, word shape, dependency role, context words, dependency parent, and most likely topic.

Embedding-based approaches to entity classification

Dong *et al.* (2015a) introduce a hybrid neural model that classifies entity mentions into a set of entity types derived from DBpedia. They introduce the notion of a mention model and obtain a vector-based representation of an entity from the words the mentions contain, estimated on automatically generated training data based on linked entity mentions in Wikipedia, similar to (Ling and Weld, 2012). Another component, the context model, obtains representations of the contextual information around a mention. The representations obtained from these two components are then utilized to predict the type distribution. The mention model is built using a recurrent neural network (RNN) architecture and the vector of an entity mention is computed from the vectors of the words in the mention. The context model is based on a multi-layer perceptron (MLP) where context words are represented as low-dimensional vectors that are different from the ones in the mention model. Context word vectors are concatenated and fed into a hidden layer that produces an l-dimensional vector. The mention model and context model are jointly trained as they are both fed to a softmax classifier that computes type assignment distributions. During training, the cross-entropy errors between the predicted and ground truth distributions are minimized, and errors are back-propagated to the two models. This neural model outperforms a strong feature-based classification approach (Ling and Weld, 2012) on a dataset constructed from news data, obtaining a $\sim$3% improvement in F_1 score without hand-crafted features and external resources.

Yogatama *et al.* (2015) propose an approach based on type embeddings that allows for information sharing among related labels. They learn an embedding for each label and each feature such that labels that frequently co-occur are close in the embedding space. In contrast with the previous approach, Yogatama *et al.* learn both instance feature vectors and type labels using a low dimensional space $\mathcal{R}^d$ such that the instance is close to its label in this space. This method also significantly outperforms (Ling and Weld, 2012) on the same Wikipedia dataset, further confirming the potential of neural methods for entity classification.

Also using embeddings, Ren *et al.* (2016a) propose an approach that is based on extracting text features for entity mentions and performing joint embedding of entity mentions M and type hierarchy T into the *same low-dimensional space* such that objects that are close in the embedding space also share similar types. They estimate a so-called type-path, that is, a path connecting multiple type assignments of a mention on the hierarchy T using the learned embeddings. This search is performed in a top-down manner, selecting the most similar types based on embeddings at every step. Ren *et al.* (2016a) introduce a novel embedding method to separately model clean and noisy mentions and incorporate a given type hierarchy to induce loss functions. They formulate a joint optimization problem to learn embeddings for mentions and type-paths and develop an iterative algorithm to solve the problem. This method turns out to be very successful, outperforming many feature-based and neural methods for entity classification, including those in (Ling and Weld, 2012; Yogatama *et al.*, 2015; Dong *et al.*, 2015a; Yosef *et al.*, 2012) on the Wikipedia dataset (Ling and Weld, 2012) in terms of both F_1 and *accuracy*. An analysis shows that this improvement is mainly achieved through modeling the type correlations and type noise.

Shimaoka *et al.* (2016) introduce a neural method for fine-grained entity classification using attention mechanism. Mention representations are obtained by averaging the embeddings of the words in the mention. Several methods to obtain context representations are compared: averaging encoder, LSTM encoder, and attentive encoder. The attentive encoder works by adding an attention layer on top of the LSTM output. In (Shimaoka *et al.*, 2017), the attentive model is expanded with constraints on the class annotations. This work also includes extensive comparisons of feature-based and neural models, and investigates how hierarchical class constraints could help in improving the performance of the models. Most neural models are based on recurrent neural networks for performance reasons. In (Strubell *et al.*, 2017), a diluted convolutional method is introduced. This model comes with the advantage of faster preprocessing than LSTM-based approaches.

Fine-grained entity mention classification systems are typically trained in a distant supervision manner, utilizing labels from knowledge

bases that might be incorrect in the local context for some mentions, because not all assigned types of an entity are relevant in the context of a sentence. Applying distant supervision in this case will result in noisy labels. Focusing on this noisy labeling problem (similar to (Gillick *et al.*, 2014)), Ren *et al.* (2016b) perform automatic identification of correct type labels for training examples, given the set of candidate type labels from a type hierarchy. This noise reduction strategy is very effective, improving the performance when applied on top of existing classification methods such as those in (Ling and Weld, 2012) by up to ∼33%.

Abhishek *et al.* (2017) address the problem of noisy training data by separating clean and noisy mentions, and incorporating a modified hinge loss function based on this two separation. The idea is that putting more weights on the clean or unambiguous mentions will help in addressing the noise. Along similar lines, Xu and Barbosa (2018) propose a method to address the problem of noisy labels in distant supervision: *out-of-context* and *overly-specific* labels. To address the out-of-context problem, they introduce a variant of cross-entropy loss function. To address overly-specific labels, they introduce hierarchical loss normalization.

Extractor-based approaches to entity classification

Extractor-based approaches to entity classification are similar to feature-based approaches but they specifically limit the possible type assignments by applying a set of extractors leveraging signals such as the patterns of explicit type mentions, specific prefixes or suffixes of a mention, verbs following an entity mention, and types of entities occurring in a similar context. Corro *et al.* (2015), for instance, introduce a system, FINET, that generates candidate types using a sequence of multiple extractors—ranging from explicitly mentioned types to implicit types—and that subsequently selects the most appropriate type using techniques from word sense disambiguation. FINET first generates a set of candidate types using multiple extractors based on patterns, mention text, verbal phrases, and related entities. After the candidates have been generated, FINET selects the most appropriate type with a Naive

Bayes classifier utilizing context features such as words in the sentence. Corro *et al.* (2015) utilize WordNet to extract the context features based on the type's gloss and its neighbors' glosses, their neighbors, and corresponding verbs. The authors later train one classifier per coarse-grained type. FINET tends to be precision-oriented due to its conservative nature of suggesting types. In addition, its performance is superior to that of a strong feature-based baseline, HYENA (Yosef *et al.*, 2012).

3.2.2 Relation of entity recognition and classification to other tasks

Entity recognition and classification are fundamental KG-related activities that enable many other downstream tasks, related to both KGs and IR. For instance, multi-word expressions recognized as entity mentions can be used not only for entity linking (Section 3.1) but also as candidates for *relation extraction* (Bach and Badaskar, 2007) (see Section 5.4). Furthermore, the entity type detected during classification is an important feature for relation extraction systems (Mintz *et al.*, 2009), as such type information can provide a signal for the likelihood of certain relationships. And the individual decisions made during entity classification can be passed for entity typing at the corpus level (see Section 5.2). In *document filtering*, some important features may be extracted first by detecting mentions of entities in the document (see Section 5.3). Having a good entity recognition system is therefore crucial to extracting the correct signals.

3.2.3 Outlook on entity recognition and classification

Approaches to entity recognition and classification started with rule-based methods which are precise but expensive to maintain. These evolved into feature-based approaches including both supervised and semi-supervised methods. Initial models relied on modeling the task as a sequence classification problem. In order to address data sparsity, specific IR techniques such as those using features from passages retrieved through pseudo-relevance feedback have proven to be effective in improving performance (Dalton *et al.*, 2011) .

The emergence of deep learning techniques has had a significant

influence on this task. One notable method within this family of approaches uses LSTM-based architectures and combines information from both annotated and unannotated corpora (Lample *et al.*, 2016).

Approaches that focus on entity classification started with feature-based methods which utilize tokens, word shapes, parts-of-speech tags, and n-gram features. In the absence of genre and domain-specific training data, these initial approaches typically use type information from Wikipedia or Freebase and train models in a distant supervision fashion (Ling and Weld, 2012). However, as each entity may have more than one type, vanilla distant supervision methods can introduce labelling noise. Follow-up methods aimed to address this issue by constraining the possible type assignments depending on the context of the mention (Gillick *et al.*, 2014).

Neural methods for entity classification begin by representing a mention model to represent the entities, and also a context model built from the words appearing close to the entity mentions. Building on this approach, subsequent methods incorporate more information on the entity types obtained from a KG, and share information between labels, i.e., entity types, to provide more signal for the classification task. They also leverage the topology of the graph and identify possible paths between entity types to further inform the classifier (Ren *et al.*, 2016b).

As distant supervision approaches are frequently employed in entity classification, a considerable amount of effort is being spent on improving over a standard form of distant supervision, i.e., simply using the labels in the KG as-is. Three main strategies have been proposed (Gillick *et al.*, 2014; Abhishek *et al.*, 2017; Xu and Barbosa, 2018). First, making sure label assignments are relevant in the context of the local mention. Another strategy aims to separate clean from noisy mentions and incorporating this separation in the training loss function. Finally, overly specific labels may be pruned in order to improve generalizability.

Although entity recognition and classification for English has achieved a good performance on popular domains such as news—e.g., achieving an F_1 score of 90.90 on the CoNLL 2003 test set (Passos *et al.*, 2014)—this level of performance does not translate to all domains or all languages. Interesting directions for future research include *domain-specific entity*

recognition and *entity recognition on lesser-resourced languages*. These directions are motivated by the fact that for building KGs we sometimes have to work within a specific domain or with lesser-resourced languages.

Sometimes there is a need to build a KG for a specific domain. In work in this direction Prokofyev *et al.* (2014) consider the task of named entity recognition for idiosyncratic document collections. Tao *et al.* (2015) focus on entity extraction in an enterprise setting, while Tang *et al.* (2015) consider the task of entity recognition and linking in a social media context. To improve the recognition performance on a specific domain, encoding more background knowledge in the recognition and classification algorithm is an unsolved challenge.

Documents in lesser-resourced languages may be the source for KG completion. For lesser-resourced languages, it would be interesting to apply transfer learning or distant supervision approaches to improve the entity recognition. One way to achieve this is by applying machine translation or a heuristic text alignment technique to generate pseudo-training data for the lesser-resourced language.

4

Knowledge Graphs for Information Retrieval

How exactly can knowledge graphs (KGs) help information retrieval (IR)? We answer this question by means of several tasks. In general, entities taken from a KG can be leveraged within an IR system in order to help improve the understanding of a user's intent, queries, and documents beyond what can be achieved through word tokens on their own. Having a KG also allows us to answer information needs that might be more amenable to be answered directly, as opposed to returning a ranked list of documents. Similarly, KGs enable the exploration of related entities mentioned in a document collection or a search engine result page. Finally, KGs can help to provide explanations of entities and relationships in context in order to further support the user. In sum, KGs allow us to enhance the user's search experience through a better understanding of intent, of queries and documents, through direct answers, and through enhanced exploration facilities.

We start our discussion with a core IR task: document retrieval. In particular, we discuss how entities detected in queries and documents can be used to improve document retrieval (Section 4.1). After that, we focus on the task of retrieving entities given a query so as to satisfy an information need (Section 4.2), and continue with recommending

related entities given an query entity (Section 4.3). To close this chapter, we discuss an emerging task: explaining relationships between entities (Section 4.4). Table 4.1 provides a structured summary of the tasks and approaches we discuss in this chapter.

Table 4.1: Structured summary of IR tasks and approaches discussed in Chapter 4.

Task and approaches	**Description**
Document retrieval (Section 4.1)	Rank documents given a query.
expansion-based	Expand queries and/or documents with entity-based information.
latent factor modeling	Model and leverage a latent space between query and documents.
language modeling	Incorporate term sequences marked as entities when building language models of a query and a document.
deep learning	Incorporate KG-based embeddings to improve query/document representations and steer the retrieval process.
Entity retrieval (Section 4.2)	Rank entities in text or KG given a query.
language modeling	Retrieve entities by matching a query with entity descriptions or mentioning documents.
neural language modeling	Learn latent representations of query and entities, compare for retrieval.
multi-fielded representation	Represent an entity as a multi-fielded document and use document retrieval techniques.
Entity recommendation (Section 4.3)	Recommend related entities given an entity and/or context.
heuristic	Estimate statistical associations between entities from text.
behavioral	Recommend entities based on similar users' interest.

graph-based	Recommend entities based on the structural connections in a graph.
Relationship explanation (Section 4.4)	Explain the relationship between a pair of entities.
instance-based	Explain the relationship by selecting a set of key related entities.
description ranking	Generate and rank candidate explanations from external corpora.

4.1 Document retrieval

Compared to the vast body of literature on document retrieval in general, and also more recent tasks such as entity linking, there is relatively little work that leverages knowledge graphs to improve document retrieval. The chief reason for this is that understanding precisely how to effectively leverage entity annotations and text in conjunction to improve ad-hoc document retrieval is as-yet unsolved. Let us first formally define document retrieval as follows:

Definition 4.1 (Document retrieval)**.** Given a query q and a collection of documents D, score and rank each document $d \in D$ based on its relevance to q.

4.1.1 Approaches to document retrieval

Approaches to document retrieval that leverage entity-oriented information from KGs can be grouped into *expansion-based*, *latent factor modeling*, *language modeling*, and *deep learning* approaches. *Expansion-based* approaches explicitly incorporate entity-oriented information as features in the retrieval process. In contrast, *latent factor* approaches do not attempt to enrich query or document representations from a KG directly, but aim to extract concepts inherent in queries and documents. *Language modeling* approaches consider semantic information when computing retrieval scores using language modeling while *deep learning* methods in this context may leverage KG-based embeddings to improve

query and document representations or change the matching function to incorporate these vectorial representations.

Expansion-based approaches to document retrieval

Some of the work on document retrieval that leverages entity-oriented information can be viewed as a variant of query expansion. E.g., Dalton *et al.* (2014) employ query expansion techniques that enrich the query with features from entities and their links to KGs, including structured attributes and text. They experiment with both explicit query annotations and latent entities and introduce the *entity query feature expansion* (EQFE) model, which works as follows:

Preprocessing First, documents are preprocessed with entity linking, and additional information obtained from knowledge graphs is indexed as different fields of the document.

Query annotation At query time, the query is also preprocessed with entity linking, providing annotations for all entity mentions in the query.

Expansion from feedback Two types of relevance feedback are then considered for expansion: (1) KG feedback, in which the query is issued against an index of a KG in order to retrieve related entities, and (2) corpus-based feedback in which related entities are obtained from retrieved documents.

The different expansion strategies include related words, entities, mentions, types, categories and neighbors. Each expansion strategy can be incorporated as a field or a representation of the document. Feature weights are learned for each of these different expansions with a log-linear learning to rank approach.

To evaluate the effectiveness of their expansion method, Dalton *et al.* consider three test collections: TREC Robust04, ClueWeb09B, and ClueWeb12B.[1] They compare entity query feature expansion (EQFE) against a sequential dependence model (SDM), SDM with collection

[1] Test collections and other resources are described in Appendix B.

relevance, and a relevance feedback model. EQFE achieves the best performance in terms of mean average precision (MAP) on Robust04; it also obtains the best performance in terms of NDCG@20, ERR@20, and MAP on the ClueWeb12B collection. On CluebWeb09B no improvement is obtained, which is interesting as ClueWeb09B is the only corpus with explicit entity annotations. The reason that the method fails to obtain any improvements on ClueWeb09B can possibly be attributed to the fact that ∼37% of relevant documents in ClueWeb09B do not contain an explicit query entity. Also, ∼73% of the documents returned by simply retrieving the entity identifiers are unjudged, which likely means that the performance of the method is substantially underestimated.

Xiong and Callan (2015b) propose a method to improve document retrieval by using knowledge graphs for query expansion. They consider two methods for performing entity-oriented query expansion: *unsupervised* and *supervised* expansion. The method consists of two main steps: (1) object linking, and (2) term selection. In the object linking step, ranked lists of related KG entities are generated. Two approaches are considered for object linking: issuing a query to the Google Search API, and selecting entities from FACC1 annotations[2] in the top-ranked documents. In the term selection, related terms from the linked objects' descripts are ranked for expansion. The unsupervised expansion approaches combined several variants of linking and term selection strategies. In constrast, their supervised method to query expansion considers three features derived from the individual method variants for ranking candidate terms for expansion. When applied on the ClueWeb09B document collection, the expansion-based method introduced by Xiong and Callan (2015b) outperforms common state-of-the-art expansion systems and also EQFE (Dalton *et al.*, 2014). Note that the differences in performance between the methods could also be attributed to the underlying entity linking method that was applied.

[2]FACC1 annotations are automatic annotations of English web pages from ClueWeb09 and ClueWeb12 to Freebase entities (Gabrilovich *et al.*, 2013).

Latent factor approaches to document retrieval

In contrast with expansion-based approaches, latent factor approaches do not attempt to enrich query or document representations from a KG directly, but aim to extract concepts inherent in queries and documents.

We start by discussing an early publication on latent factor modeling that does not use KG information, but rather extracts the latent factors from queries and documents. Metzler and Croft (2007) introduce Latent Concept Expansion, an approach based on Markov Random Fields (MRFs) which aims to discover latent concept given an original query issued by the user. Their main intuition is that a query might contain latent concepts that are directly expressed by the user. The idea is to recover these concepts by modeling the term dependencies. From a graph representation of a query G, which contains the query terms, an expanded graph H can derived by adding single and multiple terms concepts. A probability distribution of over latent concepts is inferred from a small number of relevant or pseudo-relevant documents for query q. To perform query expansion, k latent concepts with the highest likelihood are selected. A new graph G' is then constructed by expanding the original graph G with selected concepts.

Next, we move on to a latent factor approach that uses KG information. Xiong and Callan (2015a) propose a document retrieval technique based on expansion using external data in knowledge graphs; they consider entity relationships as a latent space. The proposed algorithm, EsdRank, treats vocabularies, terms, and entities from external data (i.e., entities in knowledge graph or concepts in a controlled vocabulary) as a means to connect a query and documents. One key component of the method is Latent-ListMLE, a list-wise learning to rank model. Latent-ListMLE reranks an initial set of documents with the help of related entities and feature vectors. The feature vectors are derived from the relationships between entities and documents, and another feature vector, which represents the relationship between the entity and the query. Three strategies are considered to find entities given a query and document: query annotation, entity search, and document annotation. Xiong and Callan use a feature representation that is inspired by Dalton *et al.* (2014). The relationships between query, documents, and enti-

ties are represented by a set of features that describe the relationship between query and entities (including entity selection score, textual similarity score, ontology overlap, entity frequency, etc.) as well as a set of features that describe the relationship between entities and documents (including textual similarity, ontology overlap, graph connection, and document quality). The best combination of query representation and document ranking is then learned from these features.

EsdRank outperforms EQFE (Dalton *et al.*, 2014) on the ClueWeb09B and ClueWeb12B datasets on almost all metrics. It is interesting to note that EsdRank achieves an improvement on ClueWeb09B, where EQFE fails. In addition, finding relevant entities for query and documents (i.e., entity selection) is an important step in using external data for ranking. One important takeaway from the experiments in (Xiong and Callan, 2015a) is that query annotation, i.e., entity linking on queries, is the most reliable method for selecting related objects to improve document retrieval. Instead of search or annotation-based associations between query and entities, Xiong and Callan (2015a) use an entity linking method to infer the query-object association. In their experiments, Xiong and Callan (2015a) utilize the TAGME entity linking method (Ferragina and Scaiella, 2010). When compared against a language modeling approach by Raviv *et al.* (2016), the method by Xiong and Callan (2015a) performs much better with approximately 0.20 absolute difference in MAP@100 on ClueWeb09B.

Liu and Fang (2015) introduce Latent Entity Space (LES), an approach that maps queries and documents to a high-dimensional latent entity space. Each dimension in the latent space corresponds to one entity. Similar to previous approaches in this line of work, the idea is to capture the semantic content of queries and documents better. Information around an entity in each dimension is captured by a profile of the entity. Two approaches are explored to build the entity profile: combining information around the entity across multiple documents in the corpus, and also using the entity profile from an external KGs. What is unique is that the latent space is constructed in a *query-dependent* manner. At query time, only few entities that are highly related to the query aree used in the construction of the latent space. The relevant entities for each query are obtained by

performing *query projection*, i.e., the weighted sum of the similarity between each entity in the query and each entity profile is computed with a query likelihood model. After the entities in the latent space have been selected, *document projections* into the latent entity space are computed. For each entity in the latent space, the similarity between the entity model and document model is computed using KL-divergence. The final matching step involves an interpolation between this latent entity score and the query likelihood score. The method has been evaluated on the CluebWeb09B, TREC 2013 Web Track, and TREC 2014 Web Track. When compared against EQFE, the proposed approach provides significant improvements. The improvements can be attributed to the robustness of LES against low entity annotation quality, as it does not directly use the entity annotations in the relevant documents, but rather relies on comparing the language model of the document and the entity profiles. Note that there are similarities between the query and document projection methods employed in LES to the latent layer mapping in Latent-ListMLE, a component of EsdRank (Xiong and Callan, 2015a). The main difference is that EsdRank combines the components in a supervised fashion and also uses handcrafted query-entity and document-entity features.

Language modeling approaches to document retrieval

Similarly to some of the methods detailed above, Raviv *et al.* (2016) devise an entity-based language model that uses entity linking methods. Their model takes into account the uncertainty inherent in the entity linking process and also incorporates a balance between using entity-based and term-based information. They apply entity linking to obtain entities along with the linking confidence score estimated by an entity linking method. Based on the output of this annotation, a unigram entity-based language model over a token space can be defined. The token space includes the set of all terms in the document collection and the set of entities that were linked at least once within a document. The most important concept in this model is the notion of a so-called pseudo-count that captures the uncertainty mentioned above. Two strategies are considered: hard and soft confidence thresholding. In hard thresholding,

a threshold is placed on the confidence score of each annotation and those mentions that are linked with a certain confidence score are considered for pseudo-counts. In soft thresholding, the confidence score of linking a particular mention is taken as pseudo-count during the estimation, interpolated using an importance parameter.

Raviv *et al.* (2016) perform retrieval experiments on the AP, Text Retrieval Conference (TREC) Robust04, WT10G, GOV2, and ClueWeb09B test collections. The experimental results indicate that the entity-based language model with hard and soft thresholding improves over the standard term-based language model. Raviv *et al.* also learn that their methods are robust with respect to different entity linkers.

In another language modeling approach, Ensan and Bagheri (2017) propose a document retrieval model that uses "semantic linking" on the graph representation of documents and queries. They complement keyword-based retrieval models with semantic information. Their method, Semantics Enabled Language Model (SELM), is based on the query likelihood model but instead of computing the likelihood based on terms, it is computed on an undirected graphical model built around the entities. The entities or concepts in the documents are treated as observed variables, while the entities in queries are modelled as target variables. Their experiments demonstrate that SELM can complement the performance of keyword-based systems. When interpolated with other retrieval models, this method successfully improves the performance.

As many of its components are derived from language modeling techniques, LES (Liu and Fang, 2015) can also be considered a *language modeling* approach to document retrieval. The main difference between LES and approaches in (Raviv *et al.*, 2016) and (Ensan and Bagheri, 2017) is that it does not explicitly use entity linking in its components, but rather relies on entity profiles that are estimated from a corpus or adapted from KGs.

Deep learning approaches to document retrieval

Xiong *et al.* (2017b) introduce Explicit Semantic Ranking (ESR), a ranking technique that leverages knowledge graph embeddings. ESR

represents queries and documents by embeddings of entities in the knowledge graph. Semantic relatedness between the representations of query and documents is then computed in the embedding space. Embeddings of entities are trained from edges in a knowledge graph using a skip-gram based approach. First, query and documents are represented as bags of entities using an entity linking method. Next, a translation matrix that captures the relatedness between entities in the query and documents is constructed. From this translation matrix, histogram features of entity relatedness are computed by grouping them by strength into several bins. Finally, the histogram-based features are used to rank documents with a learning to rank-based approach.

In their follow-up paper, Xiong *et al.* (2017a) consider enriching queries and documents with entity information from knowledge graphs. This approach models query and documents as word-based representations and entity-based representations simultaneously. By incorporating both types of information, they consider four types of interaction derived from words and entities in the query and documents, and subsequently use them as features for ranking. One distinguishing feature of this approach is that it employs an attention-based mechanism to address the uncertainty in the entity linking step. The method works by first building bag-of-words and bag-of-entities representations of query and documents. Then, matching features between (query words, documents words), (query entities, document words), (query words, document entities), and (query entities, document entities) are extracted based on various standard IR models. For matching (query entities, document entities) specifically, the authors follow their previous approach, ESR (Xiong *et al.*, 2017b).

Liu *et al.* (2018) introduce the Entity-Duet neural Ranking Model (EDRM), an approach to incorporate semantic information from knowledge graphs in neural ranking systems. Inspired by the improvements brought about by entity-based models to feature-based ranking systems, the authors study the impact on neural retrieval systems. This work follows the same search framework as the word-entity duet introduced previously by Xiong *et al.* (2017a), which starts by building bag-of-words and bag-of-entities representations of queries and documents. The main difference, however, is that EDRM captures the matching

between queries and documents through a translation layer in a neural architecture, instead of using handcrafted features.

Another method that follows (Xiong *et al.*, 2017a) is proposed in (Shen *et al.*, 2018), although it does not involve deep learning in the matching process. Shen *et al.* (2018)'s method follows the approach of representing query and documents as bag-of-words and bag-of-entities. It is designed for the literature search domain, where queries frequently contain multiple entities with different types. Standard, unigram bag-of-words and bag-of-entities are used to represent the documents after an entity linking step. What is unique in their approach is that the query is represented as a heterogeneous graph, where the nodes represent query tokens and the edges represent latent relations between two query tokens. For word tokens, edges are created between adjacent words. For entity tokens, type information relationships between pairs of entities are used as weights of the edges. The matching process between query and documents is solved as a graph covering process in an unsupervised fashion. The idea is to rank documents that could cover more information needs higher in the search results. Experiments on the TREC-BIO dataset demonstrate the effectiveness of the method, especially for queries containing multiple entities. The intuitions underlying this method can also be connected to earlier work on modeling latent concepts in queries based on MRFs (Metzler and Croft, 2007).

4.1.2 Relation of document retrieval to other tasks

In the context of this survey, document retrieval is closely related to *entity linking* (Section 3.1), as most approaches to document retrieval that use entity information primarily depend on performing entity linking on the queries and the candidate documents. We note that adhoc document retrieval methods are critical to some *entity retrieval* (Section 4.2) or entity recommendation (Section 4.3) tasks, in which relevant documents are first retrieved, and then entities found in these documents are ranked.

KGs also help in a setting where multilingual support is a concern. Here we discuss how knowledge graphs are currently being leveraged in this setting. For instance, Franco-Salvador *et al.* (2014) obtain a

language-independent representation of documents containing concepts and relations between them. The key concepts of a document are represented as a graph, which are later complemented with terms appearing in the document. Similarity between documents is computed from a combination of graph and document similarities. Franco-Salvador *et al.* show that using knowledge graphs helps to improve the performance on the task of comparable document retrieval, i.e., retrieving similar documents in another language.

Zhang *et al.* (2016b) present an entity-based system for multilingual and cross-lingual IR. They transform keyword queries and documents to a semantic form in order to facilitate query disambiguation and overcome the vocabulary gap. Their query understanding approach is presented in (Zhang *et al.*, 2016c). The approach works by matching keyword queries to entity graphs in the KG. Cross-lingual links between Wikipedia entities are leveraged and surface forms of entities across languages are extracted and utilized during the interpretation of keywords to entity graphs.

4.1.3 Outlook on document retrieval

Looking back at the document retrieval approaches that utilize KGs to improve document retrieval methods, we first observe that initial expansion-based approaches identified entities mentioned in the query and documents by applying either entity linking or related entity finding methods. Then, having obtained the associated entities, the next step was to incorporate information from the associated entities in various ways. Latent-space approaches focused on modeling the query and/or documents into lower-dimensional latent factor based representations, which can be applied in query-dependent or query-independent manner. With the emergence of deep learning, later approaches to document retrieval began to utilize neural methods. The most successful approaches combine both word and entity-based representations (Xiong *et al.*, 2017a; Mitra and Craswell, 2018; Onal *et al.*, 2018).

We also conclude that, regardless of the approach or direction used, an important factor in end-to-end performance is the dependence of all methods on the entity linking method that is being used. Furthermore,

several publications have shown that entity-based methods complement keyword-based methods and that a combination of the two often yields a further increase in performance (Section 4.1.1).

Document retrieval can be improved by *semantic matching*, i.e., going beyond the traditional term-based approach. One way to achieve that is by applying neural methods. Another direction is by representing the relationships between text and entities within and across documents. With the emergence of deep learning in information retrieval (Onal *et al.*, 2018), we expect more *neural entity-enhanced document retrieval* methods to emerge. One general strategy would be jointly learning the representation of documents, queries, and entities, and using those to improve document retrieval in combination with more traditional term-based methods.

Recently, pre-trained contextual word representations have proven to be effective in improving various search and natural language tasks, including retrieval (Devlin *et al.*, 2018). Combining this approach with knowledge graphs would be interesting to explore.

4.2 Entity retrieval

Entity retrieval has attracted significant attention through the launch of the expert finding track at TREC (Craswell *et al.*, 2005). Since then there have been various incarnations at different venues such as INEX (de Vries *et al.*, 2008), and also with alternative settings, e.g., ranking entities as found in document collections, in knowledge graphs, or in both. A TREC track devoted to entity retrieval has run from 2009 until 2011 (Balog *et al.*, 2009b). We define entity retrieval as follows:

Definition 4.2 (Entity retrieval)**.** Given a query q and a document collection D, retrieve and rank entities mentioned in or associated with each document $d \in D$ according to their relevance to q.

We also formalize another setup of entity retrieval below, where candidate entities are obtained from a KG.

Definition 4.3 (Entity retrieval from KG)**.** Given a query q and a knowledge graph KG, retrieve and rank entities in the KGs according to their relevance to q.

Forms of entity retrieval considered in the literature include *term-based entity retrieval*, *ad-hoc object retrieval*, and *list retrieval*. Here, we focus on the first two forms, and consider list retrieval as a specific form of entity recommendation, to be discussed in the next section.

4.2.1 Approaches to entity retrieval

We group approaches to entity retrieval into *language modeling*, *neural* approaches, which are more closely associated with the *term-based entity retrieval* setting (i.e., the first setting in our formalization) in which no explicit representations of entities are provided. The *language modeling* approaches consist of methods spawned around the task of expert finding, building on various extensions of classic language modeling methods. In contrast, *neural approaches* aim to learn distributed word representation of entities, which are optimized for retrieval. In *ad-hoc object retrieval* (i.e., the second setting in our formalization), the entities are considered as objects with attributes and relationships, hence they can be represented as multi-fielded documents. Within this setting, *multi-fielded representation* approaches, which represent entities and documents as a set of fields, will be discussed.

Language modeling approaches to entity retrieval

Language modeling approaches to entity retrieval originate from work on expert finding (Balog *et al.*, 2006; Balog *et al.*, 2009a). The authors introduce two models for ranking entities given a query with two strategies: representing an entity as a virtual document, and ranking the documents given the query mentioning certain entities.

The first strategy is called the *candidate-centric* model. The main idea is to build a textual representation of each candidate expert and then rank them based on the query using traditional retrieval models. One way of doing so is by representing a candidate expert as a multinomial distribution over a vocabulary of terms, and predicting how likely a candidate would generate the query. In the second strategy, called the *document-centric model*, one first finds documents that are relevant to the query and then identifies the experts associated with these documents (Balog *et al.*, 2012). The document-centric model is

more robust and more effective than the candidate-based model, and it is often considered as a baseline for the expert finding task as it can easily be implemented on top of an existing document retrieval system. Balog *et al.* (2011) focus on the query modeling aspects of entity retrieval; they consider terms, categories, example results as sources of information for this purpose, and demonstrate the contribution on the Initiative for the Evaluation of XML Retrieval (INEX) entity retrieval task.

Also in this line of work, Petkova and Croft (2008) introduce hierarchical language models for expert finding in enterprise corpora. In particular, they propose a query-independent approach that build term-based representation of candidate experts. One particular feature of their approach is that they model an expert as a mixture of documents, rather than one long document.

Neural approaches to entity retrieval

In the setting of expert finding, Van Gysel *et al.* (2016b) introduce an unsupervised discriminative model for the task. They learn distributed word representations in an unsupervised way, constructing them solely from textual evidence. More specifically, they learn a log-linear model of probabilities of a candidate entity given the word. In later work, Van Gysel *et al.* (2016a) improve their approach to learn term and entity representations in a different space, by adjusting the representations so that they are close in the entity space.

Van Gysel *et al.* (2016a) confirm the effectiveness of their Latent Semantic Entities (LSE) approach for retrieval when used in combination with query-independent features and the query likelihood model. LSE outperforms other latent vector space baselines (i.e., Latent Semantic Indexing (LSI), latent Dirichlet allocation (LDA), and word2vec) for lower-dimensional vector spaces. One key insight from their work is that this neural approach and term-based retrieval make very different errors. In some cases, the retrieval performance is significantly improved by the semantic matching capability provided by LSE (Van Gysel *et al.*, 2017b). An expansion of LSE, called neural vector space model (NVSM), that adds increased regularization, and accelerated training has been

proposed by Van Gysel *et al.* (2018).

Multi-fielded representations

A well-studied entity retrieval setting is the *ad-hoc object retrieval task*, which focuses on entities, their attributes, and their relationships in a KG. The goal is to retrieve a list of resource objects (i.e., entities) with respect to a user query.

Next, we discuss supervised approaches to entity retrieval in this context. One popular supervised method for entity retrieval is using *multi-fielded representations*: an entity is represented as a set of fields with bag-of-words values. The approaches within this group are tied to retrieval of semi-structured documents in general, including the work by Kim *et al.* (2009) which propose a probabilistic approach for ranking multi-fielded documents. This method relies on a mapping probability, i.e., the posterior probability that a query will be mapped to a certain field in the document. This mapping probability can be estimated by considering how often a certain term appears in a certain field, and subsequently used to weight the score computed for each field in the entity representation.

Pound *et al.* (2010) define the formalization, setting, and experimental setup for the task of ad-hoc object retrieval in an entity-based context. One simple baseline for this task in a graph-based setting is simply considering TF.IDF over the entity properties in the graph, i.e., term frequency (TF) and inverse document frequency (IDF) statistics are computed for every property of the entity in the graph. Several methods aim to learn appropriate weights for each field (Pound *et al.*, 2010).

Tonon *et al.* (2012) propose a hybrid approach that combines IR and structured search techniques. They propose an architecture that exploits an inverted index to answer keyword queries along with graph-based information to improve search effectiveness over a linked data graph. Each object in the graph is represented with the following pieces of information: entity names in URIs, entity names in labels, and attribute values of the entity. This information is indexed as a structured, multi-fielded index on top of which multi-fielded retrieval algorithms such

as BM25F can be employed. The additional benefit of having a graph structure is that additional relationship data can be used as a context to improve object retrieval. Tonon *et al.* (2012) incorporate additional methods based on query expansion and relevance feedback on the graph data, and apply these in combination with the basic BM25F ranking. The use of structured search on top of standard IR approaches can lead to significantly better results–graph-based extensions can obtain up to a 25% improvement in MAP over a retrieval-based baseline.

Similar to Tonon *et al.* (2012), Zhiltsov and Agichtein (2013) leverage relationship information to improve entity retrieval. They integrate latent semantic information to improve entity search and combine the compact representation of semantic similarity with explicit entity information. The authors represent an entity as common fields such as names, attributes, and outgoing links. In addition, the relationship between entities is incorporated by representing the entity relationship graph as a tensor. Zhiltsov and Agichtein (2013) factorize the tensor into a number of latent factors, and later enrich the fielded representations of the entity with top-related entities obtained through latent factor modeling.

Addressing the same task, Zhiltsov *et al.* (2015) adapt term dependency models, as they are known to be more effective than unigram bag-of-word models for ad-hoc document retrieval. They propose the fielded sequential dependence model (FSDM), a term dependence model for entity retrieval that is similar to the Markov Random Field model (Metzler and Croft, 2005).

Later, Nikolaev *et al.* (2016) extend this model by generalizing it: instead of learning the field weight parameters directly, the dependencies between the query terms and fields are taken into account and parameterized as a set of features based on the contribution of query concepts matched in a field towards the retrieval score. The features that are used for this parameterization are collection statistics, part-of-speech features, and proper noun features. Experimental results indicate that the parameterization helps to improve the performance over FSDM. Taking into account both term dependencies and feature-based matching of query concepts to document fields is beneficial. Parameterizing the field importance weight results in a higher number of queries that

are helped and also a greater magnitude of improvements.

Hasibi *et al.* (2016) exploit entity linking for entity retrieval. They introduce entity linking incorporated retrieval (ELR), a component that can be applied on top of any term-based entity retrieval model based on the Markov Random Field framework. They extend the Markov Random Field approach and incorporate entity annotations into the retrieval model, similar to the FSDM model introduced in (Zhiltsov *et al.*, 2015) with a term that weights the importance of entity annotations; this introduces entity-based matching in addition to term-based matching. Hasibi *et al.* (2016) evaluate the effectiveness of their approach on the DBpedia entity collection (Balog and Neumayer, 2013). They compare their approach to state-of-the-art entity retrieval methods such as SDM and FSDM (Zhiltsov *et al.*, 2015). Experimental results confirm the effectiveness of ELR when applied on top of these retrieval methods. The improvements obtained are between 6.3–7.4% in terms of MAP and 4.5–6.1% for P@10. Their results also indicate that ELR especially helps to improve the performance on complex and heterogeneous queries.

Both MRF based methods, (Hasibi *et al.*, 2016) and (Zhiltsov *et al.*, 2015), can be considered evolutions of older concept-based retrieval methods, e.g., Latent Concept Expansion introduced by Metzler and Croft (2007). It is an approach based on MRFs that aims to discover latent concepts given an original query issued by the user. Their main intuition is that a query might contain latent concepts that are directly expressed by the user. The idea is to recover these concepts by modeling the term dependencies.

Graus *et al.* (2016) propose a method for enhancing the representation of an entity from various external sources. Their method adjusts each field's importance in an online manner, learning from user interactions such as click feedback. Graus *et al.* consider the following static and dynamic description sources to build dynamic representations of entities.

- **Knowledge base**: anchor text, redirects, category titles, and titles of linked entities in a KG;
- **Web anchors**: anchor text of links to Wikipedia pages;
- **Twitter**: tweets with links to Wikipedia pages;

- **Delicious**: references to entities through social tags; and
- **Queries**: queries that can be linked to Wikipedia pages.

Entities are modeled as fielded documents where each field is a term vector that represents the entity's content from a description source. One unique feature of their approach is that the fields are updated over time. At every time point, the term vectors are updated with resources obtained from queries, tweets, and tags. Based on this dynamic representation, feature weights are learned for query-field similarity, field importance, and entity importance score based on each field and description.

In their experiments, Graus *et al.* (2016) update the fields with external representations that arrive in a streaming manner. They demonstrate that incorporating dynamic description sources into a collective entity representation allows a better matching of users' queries. They also show how continuously updating the ranker leads to improved ranking effectiveness over time.

4.2.2 Relation of entity retrieval to other tasks

Entity retrieval, if performed from text, depends on having reliable *entity recognition* and/or *entity linking* systems (see Section 3.2 and Section 3.1). Such systems are important as the candidate entities to be ranked will need to be identified from the documents first.

There are similarities between entity retrieval and *entity recommendation*, which we will discuss in Section 4.3. We distinguish between entity retrieval and recommendation as we consider recommendation a task that is more exploratory in nature than retrieval as we typically do not have an explicit query to take into account. Recently, entity retrieval has been used as a query understanding strategy analogously to entity linking to support *document retrieval* (cf. the previous sections).

4.2.3 Outlook on entity retrieval

Looking back, term-based entity retrieval from documents initially started from the work around expert finding, from which two general models emerged: (1) retrieving documents first, then extracting entities

from the retrieved documents, and (2) representing candidate entities as documents themselves, by concatenating all documents for an entity. In a variant of the second model, the candidate entities were also represented as mixtures of documents. Typically, document-centric models are more robust than candidate-centric models. Later on, neural approaches to term-based entity retrieval were explored. With the emergence of neural approaches, it was shown that they should be considered as complements to the language-modeling approaches, as these two general approaches tended to make very different errors.

As for entity retrieval from KGs, almost all methods belong to one family: multi-fielded representation, in which each entity is represented as a collection of named and nested fields (Section 4.2.1). Two types of evolution can be observed here, either in terms of *representation strategy* or in terms of *field weighting strategy*. In terms of *representation strategy*, starting from a simple TF.IDF representation over entity properties, more sophisticated representation strategies emerged later. The first evolution involves incorporating explicit relationships as a distinct type of field (Tonon *et al.*, 2012). A later direction in terms of representation incorporated implicit entity relationships, e.g., top-k related entities inferred from the KG (Zhiltsov and Agichtein, 2013). Finally, the representation strategy evolved even further to including dynamic representations that change over time, as this has been shown to improve the retrieval performance. In terms of field weighting, a basic strategy of learning field weights independently would later evolve to incorporating more sophisticated weighting strategies, including learning query-dependent field weights (Nikolaev *et al.*, 2016).

Looking forward, potential research directions on entity retrieval that we identify include two topics that involve *entity representations* as having an appropriate representation is crucial for any downstream tasks. With the emergence of alternative term-based entity representations and latent representations learned through neural methods, it is interesting to *combine both representations*. For example, one could combine the term-based collective representation introduced in (Graus *et al.*, 2016) with neural representation methods or integrate graph-based representations (Kipf and Welling, 2016) with term-based representations in a similar spirit as Ying *et al.* (2018).

Secondly, having up-to-date entity representations is important. The entity representation would need to be updated after each significant event involving the entity. One way to enrich the representation to achieve such goal is by incorporating the output of document filtering systems (see Section 5.3) or relation extraction systems (see Section 5.4).

4.3 Entity recommendation

Another entity-oriented task deals with recommending related entities in response to a textual query and a set of query entities. A well-known instantiation of this task can be found on all the major web search engines, where entities related to an entity relevant to the query are shown. We refer to this task as *entity recommendation*. In the literature, the task is sometimes also referred to as *related entity finding* (Bron *et al.*, 2010; Foley *et al.*, 2016; Kang *et al.*, 2011).

The origins of this task can be traced back to work on related entity finding, introduced at TREC 2009 (Balog *et al.*, 2009b). In this version, the expression of the information need is typically accompanied by a description of the expected related entities, and also the expected entity type. In later versions, the input is only an entity or an entity plus context words.

Definition 4.4 (Entity recommendation)**.** Given a query q (where q can be in the form of an entity or an entity plus some additional context keywords), rank each entity $e \in KG$ based on their relatedness to the query.

4.3.1 Approaches to entity recommendation

There are three general approaches to entity recommendation: *heuristic*, *behavioral*, and *graph-based*. Since the approaches that we discuss are designed for different domains and settings, they are often not directly comparable. We classify *heuristic approaches* as approaches that do not model entity recommendation as learning problems directly, but rely on statistical associations of the entities estimated from a data source. In constrast, *behavioral approaches* utilize signals derived from user interactions or feedback to generate the recommendations. Finally,

we group approaches that rely on semantic relationships without any explicit user feedback as *graph-based approaches*. We briefly discuss the performance of some of the approaches below.

Heuristic approaches to entity recommendation

Early work on ranking related entities is based on simple statistical associations. Bron *et al.* (2010) introduce a related entity ranking method utilizing simple co-occurrence statistics. They first apply and compare different co-occurrence statistics, such as Maximum Likelihood Estimation (MLE), Pointwise Mutual Information (PMI), and Log Likelihood Ratio (LLR), and later incorporate a contextual model learned from a language model of co-occurring entities as well as a type filtering model. When evaluated within the Related Entity Finding (REF) framework (Balog *et al.*, 2009b), the type filtering and context model are shown to be effective. They improve the performance of co-occurrence models by up to 115% in terms of R-precision and 29% in terms of Recall@100.

Behavioral approaches to entity recommendation

This group of approaches to entity recommendation often utilizes user feedback (e.g., clicks on related entities, documents, or entity panes) in combination with other features for recommendation. Kang *et al.* (2011) propose a machine-learned entity ranking model that leverages knowledge graphs and user data as signals to facilitate semantic search using entities. The approach jointly learns the relevance among the entities from click data and editorially assigned relevance grades. The authors use click models to generate training data to learn pairwise preferences of *entity facets*, i.e., a collection of related entities belonging to the same group. Once the facets are ranked, the related entities for a facet are ranked with feature-based models.

Continuing this line of work, Blanco *et al.* (2013) propose a learning to rank framework for entity recommendation based on various signals. They extract information from data sources such as web search logs, Twitter, and Flickr, and combine these signals with a machine learned ranking model to produce a final recommendation of entities to user

queries. The authors use features based on co-occurrence, entity popularity, and other knowledge graph features such as entity types and entity relations. Blanco *et al.* evaluate the recommendation performance by collecting judgments on the related entities output. Overall, they achieve an NDCG@5 score between 0.824–0.950 across different entity types. In addition, they find that the type of the relation is the most important feature for entity recommendation.

The next set of approaches considers the user profile and essentially estimates the conditional probability of an entity given a user profile, while other models attempt to rank entities based on context words, estimating the conditional probability of an entity given the query, or a combination of the two. Instead of recommending entities given an entity query, a contextual model incorporates information such as the profile or text currently selected/browsed by the user.

Yu *et al.* (2014a) utilize information from user click logs and knowledge extracted from Freebase. They propose heuristics and features for entity recommendation and a time-aware personalized recommendation framework to utilize the heuristics and features at different granularity levels. Their method incorporates pairwise similarity measures extracted from both user logs and the KG. It considers the consistency and the drifting nature of user interests as well as different types of entity relationships and several other heuristics. The authors include KG features such as path features, relationship features, content similarity features, and co-clicks. Most of these features are pairwise features derived from the main entity and the candidate related entity. They also incorporate point-wise features such as co-click, global popularity, current popularity, and cross-domain correlation.

Yu *et al.* (2014b) consider personalization when generating entity recommendations. They propose a graph-based approach, using a heterogeneous information network to link entities and users to generate personalized recommendations. They learn a recommendation model for each user based on the users' implicit feedback. To handle sparsity, they first discover groups of users who have similar preferences and use these groupings to learn an aggregated, personalized recommendation model. The final recommendation is generated by a combination of the user-based and group-based model.

As the setting of Yu *et al.* (2014a) and Yu *et al.* (2014b) is closer to a canonical recommendation task based on user behavior, they evaluate their approach against common baselines such as *popularity*, *co-clicks*, and *non-negative matrix factorization*. The personalized models introduced by Yu *et al.* (2014b) improve over the baseline recommendation algorithms by 12% *(co-clicks)* and 38.8% *(non-negative matrix factorization)*, respectively.

Related to the previous method that uses KGs and click log data, Bi *et al.* (2015) include a novel signal: an *entity pane* log. They propose a probabilistic entity model that provides a personalized recommendation of related entities using three data sources: *knowledge base*, *search click*, and *entity pane log*. Specifically, their model is able to extract hidden structures and capture underlying correlations among users, main entities, and related entities. Furthermore, they incorporate a clickthrough signal for popular entities, extracting three types of clickthrough rate (CTR): on related entities, on main entities and related entities, and on users, main entities, and related entities. They use feedback from the entity pane to estimate the likelihood of the data and generate training labels. The observation is a set of triples that represent clicks from a user on a related entity, and a main entity. Bi *et al.* (2015) propose a model that learns the preference between pairs of triples. Training data is created by assigning positive class labels to clicked triples, and negative class labels to non-clicked triples. Instead of learning the labels directly, the idea is to learn preferences between the clicked triples and non-clicked triples.

Lastly, Gao *et al.* (2014) utilize behavioral signals within a deep learning framework. They propose a method that observes, identifies, and detects naturally occurring signals of interestingness in click transitions between source and target documents, collected from commercial web browser logs. After identifying the keywords that represent the entities of interest to the user, they recommend other related, potentially interesting entities. They estimate interestingness by learning a mapping that quantifies the degree of interest that a user has after reading a source document. The authors train a deep semantic similarity model on web transitions and map source-target document pairs to feature vectors in a latent space such that the distance between the source document

and the corresponding target in that space is minimized. In a similar vein, Ma *et al.* (2019) propose to use neural nets to learn explainable "rules" (in the form of paths in a KG) jointly with recommendations for relevant items. Their system is able to use these rules to "explain" why a certain recommendation is made based on a combination of a KG and historic user interactions.

Graph-based approaches to entity recommendation

This group of approaches to the entity recommendation task relies primarily on the connections of entities in a graph to generate recommendations. In contrast to the methods in (Yu *et al.*, 2014a; Yu *et al.*, 2014b), the methods in this category do not use any behavioral information from users. Bordino *et al.* (2013b) explore the entity recommendation problem by focusing on serendipity aspects of recommendation. The authors examine what makes a result serendipitous by exploring the potential of entities extracted from two sources of user-generated content: Wikipedia and Yahoo! Answers. They extract entity networks from each dataset by first extracting a set of entities and then constructing an entity network by using a content-based similarity measure to create links between entities. To generate recommendations, Bordino *et al.* (2013b) employ a method based on random walks with restart to the input entity, biasing the random walk to obtain other entities directly or indirectly related to the input.

Also within the random walk framework, Bordino *et al.* (2013a) consider the task of entity-oriented query recommendation given a web page that a user is currently visiting. First, they represent a page by the set of Wikipedia entities mentioned in it. To obtain query recommendations, they propose the *entity-query graph*, which contains the entities, queries, and transitions between entities, queries, and from entities to queries. They run a Personalized PageRank computation on this graph to expand the set of entities extracted from a page, and to associate these entities with relevant query recommendations. As their goal is to recommend interesting, non-obvious connections, they introduce another metric, SRDP, as the fraction of unexpected results in the entity recommendation.

In another variant of the task, Lee *et al.* (2015) explore another entity recommendation setting for users who are reading a particular document. They present a contextual entity recommendation approach for retrieving contextually relevant entities given what the user is currently browsing. Contexts such as user text selection and the document currently being browsed by the user are incorporated as input for recommendation. An undirected graph of entities is created by including an edge between entities if there is a link to an entity on the Wikipedia page of a source entity or vice versa. For recommendation, Lee *et al.* (2015) create a subgraph containing the user-selected entity, entities in the document, and a set of candidate entities. Finally, they rank candidate entities by combining their betweenness and Personalized PageRank scores.

Similar to the work by Lee *et al.* (2015), Fuxman (2015) deals with entity recommendation given that a user is currently reading a particular article and selects a portion of the text in an article. Fuxman (2015) propoose a method that first identifies a set of candidate references. Then, it learns a prediction function to score each candidate given a text selection, the full content of the document, and the set of candidate documents. Lastly, Fuxman (2015) recommends a candidate concept if the score is above a threshold. For learning, he utilizes the multiple additive regression trees (MART) algorithm with features derived from three criteria: *context coherence*, *selection clarity*, and *reference relevance*.

Lee *et al.* (2015) and Fuxman (2015) evaluate their recommendation method in the context of Wikipedia entity recommendation. They select paragraphs from a selection of Wikipedia articles and ask crowd annotators to select phrases in the paragraphs about which they want to learn more, thus arriving at pairs of user selection (i.e., paragraphs) and context (the entities/phrases). Lee *et al.* (2015) consider baselines based on Wikipedia distance, entity retrieval and pseudo-relevance feedback in their experiments. They demonstrate that their method consistently outperforms all baselines in terms of MAP. In addition, they also discover that IR-based baselines (which construct queries from the selected phrases) and their proposed KG-based method tend to retrieve different types of entities for recommendation, making a case

for combining them. Finally, they show that using a combination of a random walk and context successfully improves the performance over using either on their own.

4.3.2 Relation of entity recommendation to other tasks

Entity recommendations have a strong connection to *entity retrieval* (Section 4.2). The main distinction is that in retrieval we are retrieving entities relevant to a query, i.e., to obtain an answer to the query; in recommendation we recommend entities related to another entity through semantic or behavioral connections.

In addition, *entity linking* (see Section 3.1) also forms a building block of entity recommendation. For example, in (Odijk *et al.*, 2015) the task of related content finding can be considered as a form of recommendation. Specifically, the task is finding video content related to a live television broadcast, leveraging the textual stream of subtitles associated with the broadcast. The query for recommendation is obtained by linking entities in the subtitles of the video.

4.3.3 Outlook on entity recommendation

Looking back, methods to generate entity recommendations started from simple, statistical methods mainly involving co-occurrence counts between entities. These methods show that type-based filtering and context words around the co-occurrences are effective in improving the performance of the baseline statistical methods. As entity recommendation began to be considered in a web setting, more sophisticated supervised methods were developed. For these supervised methods, *behavioral signals* from user activities were considered in addition to common entity features. The behavioral signals include editorial annotations, search mentions, and search interactions, not only on web search results but later on also on entity/knowledge cards that are typically shown alongside search results. Finally, the increasing popularity of neural methods brought out recommendation methods that incorporate deep learning. One strategy in this direction uses a neural network to learn transition patterns from web logs, and leverages the learned patterns to generate recommendations (Gao *et al.*, 2014). A more recent

variant of this neural method jointly generates recommendations and explainable rules that can explain the recommendations (Ma *et al.*, 2019).

In a different vein, *graph-based* methods do not consider behavioral signals from users, but rely on the graph topologies to generate recommendations. Random walks are a strategy that is frequently employed to generate entity recommendations, with as main variations the construction of the graph on which the random walk is computed as well as the way the initial weights are initialized. A random walk can be performed on many variants of an entity-based graph including co-occurrence graphs, semantic similarity graphs, query-flow graphs, etc. To complement the random walk method, context-based retrieval methods were also considered. Recent work confirmed the hypothesis that random walk-based baselines tend to retrieve different types of entities in comparison to the other, context-based retrieval methods, and that combining the two improved effectiveness even more (Section 4.3.1).

Another main conclusion revolves around metrics. All previous experiments show that for entity recommendation, the quality can be judged along many dimensions including *serendipity*, i.e., how surprising the recommendation is, and *interestingness*, i.e., how likely a user will click on or select the recommended entity.

Moving forward, interesting research directions for entity recommendation include: *encouraging explorative behavior*, *leveraging heterogeneous information*, and *incorporating context-specific recommendations*.

One important goal of entity recomendation is to support exploratory search. In follow-up work to (Blanco *et al.*, 2013), Miliaraki and Blanco (2015) conduct an in-depth analysis on how users interact with the entity recommendation system. They study users, queries, and sessions that appear to characterize explorative behavior. Taking this idea one step further would be to develop entity recommendation systems that enhance serendipity once such explorative behavior is detected.

Various types of resource are available to support the task of generating recommendations; combining and leveraging them in an effective way is not trivial. Zhang *et al.* (2016a) propose an approach to leverage heterogeneous information in a knowledge graph to improve the quality of recommender systems with neural methods. They adopt TransR (see

Section 5.4) to extract an item's structural representations by considering the heterogeneity of both entities and relationships. Besides this representation learning method, heterogeneous information encoded as graphs can also be leveraged by designing recommendation algorithms that rely solely on the semantics of the connections. In this case, learning an effective way of understanding the chain of different relationship types in the path connecting two entities would be the main challenge.

Besides purely exploratory purposes, users may have a specific recommendation goal or context when using entity recommender systems. Formalizing these goals and translating them into objective functions that can be optimized in the context of recommendation is an interesting challenge.

4.4 Entity relationship explanation

Entity relationship explanation is a relatively new, emerging task. Fang *et al.* (2011) first introduce the task. The main motivation is that explanations are required to describe entity pairs, paths between entities, as well as other relationships observed in, e.g., query logs (Ma *et al.*, 2019; Reinanda *et al.*, 2015).

Definition 4.5 (Entity relationship explanation)**.** Given a pair of entities e and e', provide an explanation, i.e., a textual description, supported by a KG, of how the pair of entities is related.

4.4.1 Approaches to entity relationship explanation

Two paradigms for generating explanations exist: *instance-based explanations* and *description ranking*. We discuss these two paradigms and the approaches belonging to them in the following section. *Instance-based explanations* aims to provide an explanation of a relationsip by returning a set of related entities. In contrast, approaches following the *description ranking* paradigm will come up with candidate textual descriptions of a relationship and provides a ranking of possible explanations.

Instance-based entity relationship explanations

Fang *et al.* (2011) focus on explaining connections between entities by utilizing KGs. They mine relationship explanation patterns, which are modeled as a graph structure, and generate an explanation instance from this pattern. Their approach consists of two main components: explanation enumeration and explanation ranking. In the enumeration phase, they generate all path instances of a specified length found in a knowledge graph. This enumeration step results in a number of paths, which are combined to form a minimal explanation. Fang *et al.* (2011) propose two kinds of interestingness measure that can be computed from the candidate paths: structure-based measures and aggregate measures. The idea is to estimate the rarity of an explanation based on these measures. The explanation candidates are ranked by any of the previous individual measures. Fang *et al.* (2011) evaluate their approach by separately evaluating the two components of their explanation system. Their explanation enumeration algorithm is evaluated by focusing on efficiency, i.e., the speed of enumerating possible explanations. As the output of their explanation method is a set of objects, the results are evaluated similar to an entity recommendation system. They ask users to judge the most interesting explanations for some entity pairs, and compute a discounted cumulative gain (DCG)-like score from this judgments.

Seufert *et al.* (2016) propose a similar approach to work on entity sets. Their method focuses on explaining the connection between two entity sets based on the concept of a so-called *relatedness core*: a dense subgraph that has strong relations with both entity query sets. Such a dense subgraph is expected to represent key events involving the entities in the sets. It is meant to find multiple sub-structures in the knowledge graph that are highly informative. The approach relies on two phases: finding relationship centers and expanding the relationship centers into a relatedness core. Relationship centers are intermediate entities that play an important role in the relationship. These centers must be connected to both query sets. They are identified by performing random walks over the graph, adapted from the center-piece subgraph (CPS) method (Tong and Faloutsos, 2006). Once the relationship centers are identified,

the subgraph is expanded to obtain the relationship core. The evaluation setup is similar to that of Fang *et al.* (2011). Explanations in the form of graph output are assessed by human annotators. Pairs of subgraphs are presented to the annotators, who are tasked to give their preferences. For the relationship explanation task, the method introduced by Seufert *et al.* (2016) outperforms the baseline CPS method and also an extension of (Fang *et al.*, 2011) for multiple query entities.

Entity relationship explanations based on ranking descriptions

Voskarides *et al.* (2015) study the problem of explaining relationships between pairs of knowledge graph entities, but aim to do so with human-readable descriptions. They extract and enrich sentences that refer to an entity pair, then rank the sentences according to how well they describe the relationship between the entities. They model the task as a learning to rank problem for sentences and employ a rich set of features, instead of individual interestingness measures as proposed in (Fang *et al.*, 2011). The approach introduced by Voskarides *et al.* (2015) requires a document collection containing the entities. They split entities' Wikipedia articles into sentences and extract sentences as candidates if they contain the surface form of the other entities, or sentences containing both entities' surface forms or links. To make candidate sentences readable outside the article, they enrich the sentence through pronoun resolution and linking. Candidate explanation sentences are then ranked by how well they describe a relationship of interest between entities. To combine various signals, each sentence is represented as features and a learning to rank approach is employed. The groups of features considered are the following: text, entity, relationship, and source features. A Random Forest classifier is then used to learn a ranking model. For evaluation, candidate sentences are judged using four relevance grades. Ranking-based metrics such as NDCG and expected reciprocal rank (ERR) are then computed on the description ranking using these judgments. The best variant of the model introduced by Voskarides *et al.* (2015) achieves an NDCG@10 score of 0.780 and an ERR@10 score of 0.378.

4.4.2 Relation of entity relationship explanation to other tasks

Relation explanation is important in the context of *entity recommendation* (see Section 4.3), as explanations allow users to understand the output of entity recommendation models better and help to discover interesting patterns of association. As for dependencies, relation explanation methods that rank external text descriptions rely on having entity recognition and classification (Section 3.2), and/or entity linking (Section 3.1) performed on the text. Sentence enrichment, as performed in (Voskarides *et al.*, 2015) requires the aforementioned components.

4.4.3 Outlook on entity relationship explanation

The area of entity relationship explanation is an emerging research area; there is no consensus yet on what the preferred entity relationship explanation is. Current proposals are quite varied, both in terms of settings and approaches considered. Approaches to instance-based entity explanation started with a focal pair of entities. Paths between the pairs of entities in the KG are first identified, and then later generated and ranked. Later on, the scope was extended to provide explanations for pairs of entity sets, which would require a different strategy. For description-based explanations, a document collection which contains the entities to be explained is required. As sentences are the typical units of explanation, effective methods for pronoun resolution and linking are needed in order to make sure that a sentence extracted from the corpus is understandable outside the context of the document (Voskarides *et al.*, 2015).

We identify two interesting directions that can be pursued in the context of relationship explanation. First, the task can be extended beyond providing explanations for adhoc pairs of entities to encompass providing explanations for *(in)directly related entities* or for *group(s) of entities*. Furthermore, *more complex explanation approaches* (including neural sequence-to-sequence models) have not received any attention in the literature so far. As textual descriptions that are tightly coupled with the entities under consideration are not always available, such neural models could also be employed for natural language generation of explanations in a transfer learning setting.

4.5 Conclusion

To recap, in this chapter we have described how KGs can help IR by discussing several entity-centric IR tasks and the role of KGs in each. Specifically, we discussed entity linking, document retrieval, entity retrieval, entity recommendation, and relationship explanation. In Chapter 5, we go in the opposite direction and detail how IR techniques can be applied for KG-related tasks, including KG construction and completion.

5

Information Retrieval for Knowledge Graphs

Reversing the focus of the previous chapter, we now consider how techniques from information retrieval (IR) can help when constructing or interacting with KGs. We consider typical KG-related tasks and show how IR techniques, methods, and methodologies can play a pivotal role in them. For instance, text classification and trend detection techniques from IR are useful for discovering novel entities. Another example is document filtering, a fundamental IR task that is important for KG construction and completion as it allows one to select items that are relevant to particular entities in a stream of documents. The filtered documents can then be fed to a relation extraction system to extract novel facts and triples for the entities. Finally, methods for estimating the quality of individual KG statements may be based on authority and classification models that have analogues in IR, e.g., in the form of spam detection and document or web page authority.

In the remainder of this chapter, we further zoom in on how IR approaches can be used for creating, improving, and updating KGs. Table 5.1 presents a structured summary of KG-related tasks and approaches; we will discuss each of these items in detail in the upcoming sections. We start our discussion with the issue of KG construction and

completion, starting with discovering entities (Section 5.1) and assigning entity types (Section 5.2), filtering relevant documents for entities (Section 5.3), and concluding with extracting relationships between entities (Section 5.4). Finally, we discuss paradigms and approaches for estimating the quality of a KG (Section 5.5).

Table 5.1: Structured summary of KG-related tasks and approaches discussed in Chapter 5.

Task and approaches	**Description**
Entity discovery (Section 5.1)	Decide whether an entity should be added as a new entry to a KG.
linking-based	Utilize the confidence score from an entity linking system to detect unlinkable entities.
feature-based	Train a classifier based on features from timestamp and text features in a supervised on semi-supervised fashion.
expansion-based	Discover new entities similar to a number of seed entities.
Entity typing (Section 5.2)	Decide which type should be assigned to an entity.
constraint-based	Define a set of class constraints and optimize through, e.g., integer linear programming.
embedding-based	Learn the association between an entity and a type embedding.
graph-based	Represent entities' associations with other entities, type context descriptions, and entity descriptions as a graph.
generative	Build a co-occurrence dictionary of entities and context nouns using translation and generation probabilities.
Document filtering (Section 5.3)	Decide whether a document contains important information about an entity.
entity-dependent	Learn a model for every entity based on lexical and distributional features.

entity-independent	Learn a single model for all entities based on distributional features.
Relation extraction (Section 5.4)	Extract entity relationships from text.
supervised, feature-based	Extract features based on a context within sentences and use supervised machine learning.
distantly-supervised, feature-based	Extract features based on relation context aggregated from multiple sentences, learn in distantly-supervised fashion.
semi-supervised pattern extraction	Learn and apply relation patterns in a semi-supervised fashion.
Link prediction (Section 5.4)	Predict new entity relations given existing relations.
latent feature models	Learn latent features of entities that explain observable facts and apply to new entities.
graph feature models	Predict new edges by learning features from the observed edges in the graph.
KG correctness estimation (Section 5.5)	Estimate the quality of set of facts in the KG.
sampling-based	Predict overall KG accuracy by sampling the facts efficiently.
Triple correctness prediction (Section 5.5)	Estimate the likelihood of a predicted KG statement.
fusion-based	Predict triple correctness by aggregating predictions of individual extractors.
Contribution quality estimation (Section 5.5)	Predict the quality of (parts of an) KG item.
feature-based	Predict contribution quality based on user contribution history, relation difficulty, and user contribution expertise.

graph-based	Leverage the graph connecting profiles and editors to estimate the quality of contributions.
Vandalism detection (Section 5.5)	Predict whether an edit in a KG is malicious.
feature-based	Predict vandalism based on content and context features.

5.1 Entity discovery

The set of entities in a knowledge graph tends to evolve as new entities emerge over time. To keep up with entities in the real world we therefore need to continuously discover emerging entities in news and other streams (Graus *et al.*, 2018). Entity discovery originated as a subtask of TAC-KBP, an evaluation campaign on entity linking and relation extraction (Ellis *et al.*, 2014). We define the task as follows:

Definition 5.1 (Entity discovery)**.** Given a set of documents D and a knowledge graph KG, detect E, i.e., a set of new entities that should be added to KG.

5.1.1 Approaches to entity discovery

Approaches to entity discovery are either *linking-based*, *feature-based*, or *expansion-based*. Linking-based approaches rely on entity linking techniques, and discover new entities based on the confidence during linking. Approaches that do not aim to solve the discovery of novel entities specifically are grouped under linking-based approaches. Feature-based approaches treat entity discovery as a prediction problem based on features extracted around the candidate entities. Finally, expansion-based approaches start with a seed of entities for each type that needs to be populated, and try to extract similar entities.

Linking and feature-based approaches are typically evaluated by annotating emerging entities found in a stream of documents such as news. Expansion-based approaches on the other hand focus on completing a seed set of entities based on the properties and other relationships

that the seed entities have in common. These approaches are typically evaluated in a batch setting and do not strictly require a document corpus. Hoffart *et al.* (2014) introduce a dataset for entity discovery with a news-based evaluation framework and propose a method that is commonly used as a baseline for this task.

Linking-based approaches to entity discovery

Originally, approaches such as those proposed by Kulkarni *et al.* (2009) utilize a global threshold to recognize entities not found in the knowledge base by an entity linking method. Entity linking systems, when attempting to link entity mentions in a text segment to KG entities, often generate confidence scores in the linking process. Early approaches to entity discovery, such as those in (Bunescu and Pasca, 2006), extract candidates for emerging entities from out-of-KG entities, i.e., ones with low scores with respect to a disambiguation score (e.g., the similarity between the context of a mention and the KB article of the entity). The assumption here is that such out-of-KG entities typically occur in similar contexts as known entities of a certain type. One particular limitation of these approaches is that finding a reliable threshold for novel entities in real-world applications might be impractical.

Feature-based approaches to entity discovery

Related to the linking-based approaches, Lin *et al.* (2012) attempt to solve the problem of detecting new entities based on their usage characteristics in a corpus over time. Their approach performs classification of unlinkable text segments. Here, they define an unlinkable text segment to be a noun phrase that cannot be linked to Wikipedia. They rely on the intuition that entities have different usage characteristics over time than non-entities, defining an entity as a noun phrase that could have a Wikipedia-style article if there were no notability or novelty considerations. They address the task by training a classifier with features primarily derived from a time-stamped document collection, leveraging various entity usage statistics from this longitudinal corpus. This base feature set is augmented by word features of the noun phrases such as capitalization and numeric modifiers. To evaluate their approach, two

annotators labeled 250 unlinked bigrams (based on the aforementioned definition) extracted from OpenIE (Banko *et al.*, 2007) assertions as *entity*, *non-entity*, or *unclear*.[1] When using the full feature set, Lin *et al.* (2012) manage to classify 78.4% of the bigrams correctly, outperforming a named entity recognition baseline.

Hoffart *et al.* (2014) focus on the most difficult case where the names of the new entities are ambiguous, i.e., the case where a mention can refer to not only new but also known entities. To address this challenge, they propose a method that measures the confidence of mapping an ambiguous mention to an existing entity. They propose a model for representing an ambiguous new entity as a set of weighted keyphrases. They extract descriptive keyphrases of a candidate emerging entity and compute the set difference of those keyphrases and entities already covered in the KG. They then cluster different mentions with similar keyphrases as a new emerging entity. To assess the effectiveness of their approach, Hoffart *et al.* (2014) introduce the AIDA-EE dataset for emerging entity discovery and compare the performance of the proposed entity discovery method against baselines based on entity linking, i.e., the Illinois Wikifier (Ratinov *et al.*, 2011) and AIDA (Hoffart *et al.*, 2011). The approach introduced by Ratinov *et al.* (2011) is an extension of a linking-based approach that takes into account additional features for novel entities.

Wu *et al.* (2016b) propose an approach to learn a novel entity classifier by modeling mention and entity representations into multiple feature spaces. They incorporate features based on contextual, topical, lexical, neural embedding, and query spaces. Contextual features include supportive entities, alien entities, and dependent words. They leverage a notion of semantic relatedness between an entity and a mention that is computed based on the relatedness of the entity in each embedding space. Within the query space, context words found in users' search history surrounding the entities are included. All of these features are combined to train a classifier based on gradient boosting trees. The proposed approach outperforms two strong baselines (Ratinov

[1] OpenIE is an unsupervised relation extraction system that returns a set of triples in the perform of (subject, verb, object); subjects and objects in these assertions yield entity candidates that can be used for downstream applications such as this.

et al., 2011; Hoffart *et al.*, 2014) on the AIDA-EE dataset, achieving 98.31% precision and 73.27% recall for the entity discovery task. In a feature ablation experiment, it is demonstrated that all feature spaces that are introduced in (Wu *et al.*, 2016b) contribute substantially to the performance, with contextual and topical features being the most important ones.

Other approaches focus on a social media setting. Graus *et al.* (2014) present a distant supervision method for generating pseudo-training data for recognizing new entities that will become entities in a knowledge graph. Their main focus is on Twitter and an entity linking system is applied to identify candidate entities in each tweet in a stream of documents. A tweet is then pooled as a candidate training example and a named entity recognition system is trained on this automatically generated ground truth. The resulting NER model trained in this manner will then be used as a new entity detector. Graus *et al.* hypothesize that new entities that should be included in the KG occur in similar contexts as entities that are already in the KG. To sample tweets to be used for training, they rely on features such as the number of mentions, number of URLs, average token length, density, and length. Their method achieves 45.99% precision and 29.69% recall when detecting new entities in tweets on an evaluation setup adjusted for this social medias setting. The gold standard was derived by subsampling Wikipedia entities and finding how many new entities are discovered with the proposed approach.

Expansion-based approaches to entity discovery

Expansion-based approaches to entity discovery leverage existing type and attribute information found in KGs and discover new entities with similar attributes. Expansion-based approaches are evaluated by a different paradigm than the previous linking-based and feature-based approaches. Here, the goal is to discover more entities within a specific category.

Early work by Sarmento *et al.* (2007) uses a co-occurrence method based on Wikipedia to expand a set of entities. Later, Bing *et al.* (2013) develop a framework for entity expansion and attribute extraction

from the web. That is, they discover new entities and extract specific attributes of these new entities. They take existing entities from a particular Wikipedia category as a seed set and explore their attribute infoboxes to obtain clues for the discovery of more entities belonging to the same category. Bing *et al.* (2013) also aim to find out the attribute value of these newly discovered entities. The framework introduced by Bing *et al.* (2013) leverages IR techniques by using the clues from Wikidata infoboxes for constructing a query to retrieve web pages that might contain new entities belonging the same category as the seed entities. The retrieved documents are then considered as candidates for entity extraction, with further processing to detect relevant segments in the pages.

The discovery of new entities within a given category is formulated as a ranking problem. The gold standard is obtained by retrieving the entities belonging to a number Wikipedia categories. Their approach achieves a P@50 score of 0.77 in this setting, outperforming two baselines: a sequence classification model based on semi-markov CRF (Sarawagi and Cohen, 2004) and another web-based extraction approach, SEAL (Wang and Cohen, 2007). The improvements can be explained as follows. First, issues with inconsistent contexts around the mentions of seed entities in web pages are addressed by generalizing patterns around the mentions. Second, text regions containing semi-structured data on the web page that is being considered for extraction are detected beforehand, which helps to eliminate noise encountered in pattern-based extraction.

5.1.2 Relation of entity discovery to other tasks

As we have discussed earlier, entity discovery can be performed alongside *entity linking*, utilizing linking confidence scores. We have discussed entity linking in detail in Section 3.1. In the next section on *entity typing* (Section 5.2), we discuss how a specific type of such attributes, i.e., the entity type, can be extracted using entity discovery clues in a corpus.

Document filtering systems (Section 5.3) can be used to automatically build an initial profile for a new entity identified through entity

discovery. Document filtering approaches focusing on the long-tail are especially useful in this case: as we discover new entities, we need to extract a set of attributes or generate a short description about them before including them in a KG.

5.1.3 Outlook on entity discovery

In the beginning, entity discovery was treated as an extension to entity linking tasks, i.e., a new entity would be considered depending on the entity linking score. However, one distinct challenge here is that different methods are calibrated differently, and therefore finding a reliable threshold that could work for entity discovery is not trivial. As things evolved, entity discovery began to be treated as a dedicated task, initially addressed using feature-based supervised classification approaches. The usage of entities over time was shown to be a useful signal for discovery, and newly discovered entities can be ambiguous, i.e., having the same or a similar surface form as already known entities, which would make the task of discovery even more challenging. To address this issue, key phrases associated with the entities were used to further distinguish them from known entities with the same surface form. Currently, state-of-art-methods for entity discovery employ contextual, topical, lexical, and embedding features (Section 5.1.1). An orthogonal, semi-supervised strategy relies on using attributes in Wikipedia infoboxes for the purpose of retrieving web pages containing similar, but new entities (Bing *et al.*, 2013).

We highlight one type of future work related to entity discovery: *automatically generating a description of newly discovered entities*. This is important because once we detect new entities, we will need to populate their relationships and, possibly, a summary description. In addition, having such descriptions will be useful in supporting other tasks, e.g., retrieval and filtering. To extract entity descriptions for newly-discovered entities, Hoffart *et al.* (2016) develop a simple approach for initializing descriptions of emerging entities in a user-friendly manner. They refine the method in (Singh *et al.*, 2016), requiring the user to provide a minimal description of an entity in the form of a name and initial keyphrases. Both approaches rely on having a human in the loop.

Fissaha Adafre and Rijke (2007) propose a semi-supervised method that depends on having sufficiently many "similar" entities as examples; it would be interesting to explore *purely automatic approaches* to address this problem.

5.2 Entity typing

Related to the problem of discovering new entities is deciding to which entity type(s) they belong. In contrast to the entity recognition and classification problem (which decides entity types for mentions), here we need to decide the relevant type(s) of an *entity*. We formally define the task as follows:

Definition 5.2 (Entity typing)**.** Given a KG and a set of documents D_e mentioning an entity e, decide whether type $t \in T$ should be assigned to annotate entity e, where T is a type system in the knowledge graph. The entity type assignment could be a hard, binary assignment or a soft assignment with a relevance score.

5.2.1 Approaches to entity typing

When new entities are detected, they may be classified into relevant entity types. Typical approaches that address this task leverage evidence from a document corpus or from similar entities in the KG; we classify the approaches into *graph-based*, *constraint-based*, *embedding-based*, or *generative*.

We characterize the approaches as follows. Graph-based approaches model the relationship between mention, entities, and types as relationships in a graph, and perform inference based on the constructed graphs. Constraint-based approaches define a set of constraints that need to be satisfied, and solve the optimization problem to decide the entity type assignments. Embedding-based methods use contextual features to learn embeddings of entities and mentions, and use the representations to decide on the typing. Finally, generative approaches model the assignment of entities, mentions, and types as a generative process, and learn the parameters of the generative model.

For the evaluation of this task, a sample of Freebase triples or entities that occur in the ClueWeb12 Corpus are commonly used with the Freebase type systems. "No Noun Phrase Left Behind" (NNPLB), a graph-based method for entity typing introduced in (Lin *et al.*, 2012), is used by many as a baseline. We will begin by introducing its main characteristics and features below.

Graph-based approaches to entity typing

Utilizing a graph-based approach, Lin *et al.* (2012), address the entity typing problem with a model that propagates class labels from labeled to unlabeled instances. This method works by finding similar entities that share the same textual relations with a target entity. Then, the types of the target entity are predicted from the types of the related entities. For evaluation, Lin *et al.* sample a number of head and tail entities from linked Freebase entities. They find that entities that share more textual relations are more likely to share the same type assignments.

Mohapatra *et al.* (2013) present a joint bootstrapping approach for entity linking and typing. Specifically, they present a bipartite graphical model for joint type-mention inference. Their typing approach is based on building models of contexts referring to types; it relies on three signals: "entity neighborhood," "language model," and "neighborhood match with snippet." The entity neighborhood signal leverages the direct or indirect information of type information from known parts of the knowledge graph, that is, it infers an entity's type based on types of related entities. The language model utilizes mention contexts from Wikipedia annotated text. Finally, the last signal utilizes the linked related entities in context. Mohapatra *et al.*'s inference approach is based on a graph-based method with maximum a posteriori labeling, a collective inference approach. They model the joint probability of an assignment of entity mention to a type and entity. Their graph-based method achieves ∼80% accuracy for classifying entity types on the entities found in ClueWeb12 corpus. The gold standard is obtained from the YAGO type assignments of Freebase entities; a sample set of entity-type pairs are sent to professional annotators.

Constraint-based approaches to entity typing

Nakashole *et al.* (2013) consider the task of both discovering and semantically typing newly emerging out-of-KB entities. Their method is based on probabilistic models that use type signatures of relational phrases and type correlation or disjointness constraints. Their solution leverages a repository of relation patterns that are organized in a type signature taxonomy. The candidate types to be assigned to an entity are determined based on the entity's co-occurrence with a type relational pattern. Their method starts by generating a number of confidence-weighted candidate types for entity e. The compatible subsets of candidate entities for an entity e are decided with an integer linear programming (ILP) formulation. The constraint is that some types are mutually exclusive. Here, the goal of the ILP is to maximize entity-type assignment weights so that known disjoint types do not get assigned together for the same entity. The types of emerging entities are collected from news data and evaluated through crowdsourcing. Their best method achieves 77%–88% precision for detecting the types of news entities, outperforming NNPLB, which achieves a precision of 46%–68%. The improvements are mainly obtained by considering the class disjointness constraint, as NNPLB sometimes assigns negatively correlated types to the same entity.

Similar to Nakashole *et al.* (2013), Dalvi *et al.* (2016) also present a method that employs class constraints imposed by an ontology. They consider two kinds of type constraint on the class hierarchy: *subset* and *mutual exclusion*. These constraints are incorporated within a mixed integer program (MIP) approach to estimate type assignments. The main difference is that Dalvi *et al.*'s method aims to infer and discover incomplete class hierarchies. As it works on an inferred type system, its performance is not directly comparable to the entity typing approaches that we discussed earlier.

Embedding-based approaches to entity typing

Embedding-based approaches to entity typing learn classifiers over sparse high-dimensional feature spaces that result from the conjunctive features of the entity mention and its context of occurrence. Yaghoob-

zadeh and Schutze (2015) propose to combine a global model with a context (i.e., mention-level) model. The global model aggregates contextual information about an entity from the corpus and then performs classification for each possible candidate type. The context model makes decisions on each occurrence of an entity within a context, irrespective of whether it expresses a certain type or not. Their global model utilizes an entity embedding $\vec{v}(e)$ and makes class predictions based on a neural approach, obtained by replacing Wikipedia anchors with entities, and learning the embedding from the contexts obtained in this manner.

Evaluated on a subset of the ClueWeb12 corpus, this approach achieves an F_1 score of 0.545 for entity typing, a substantial improvement over their re-implementation of the method introduced in (Lin *et al.*, 2012), which only achieves an F_1-score of 0.092 on the ClueWeb12-based dataset.

Generative approaches to entity typing

Bast *et al.* (2015) propose a method for assigning relevance scores for entity type assignments for people entities in particular. Their method makes use of existing facts in a distantly supervised fashion. They generate pseudo-training data by assigning entities with solely the given type or any subclass of it and negative examples based on entities that do not belong to that particular type in the KG. Bast *et al.* associate all words that co-occur with a linked mention of an entity within the same semantic context. The authors define a semantic context as a subsequence of the sentences that expresses one fact from the sentence. They consider three algorithms: binary classification based on the associated context words, counting profession words, and a generative model similar to LDA. The type distribution is later computed from the maximum likelihood estimate obtained by applying an expectation maximization procedure to infer the latent variable. For evaluation, Bast *et al.* create their own dataset based on Freebase triples; their best method achieves ∼80% accuracy on this dataset; a baseline method using random assignments achieves 55% accuracy.

5.2.2 Relation of entity typing to other tasks

Entity typing is related to fine-grained *entity classification* (see Section 3.2) at the mention level: local mention classification obtained through NER can be incorporated to influence a global decision on entity typing. Individual decisions at the mention level can be aggregated to obtain a type distribution of the entity.

As we have discussed earlier, entity typing and entity discovery (Section 5.1) complement each other in enriching the repository of entities in a KG. Finally, entity typing can be considered as a form of *KG completion*, as it enriches a KG by adding new entities.

5.2.3 Outlook on entity typing

One of the main motivations behind graph-based approaches to entity typing was that the type of an entity can be predicted from the types of entities related to it. In addition, entities that share the same set of relationship types are more likely to share the same type assignments. With the emergence of constraint-based approaches, it was shown that other types of relationship constraints are also effective in improving prediction performance. These constraints can be grouped into two sets: one based on *mutual exclusion*, i.e., one type is excluded from being assigned together with another type, and the second based on *subsets*, i.e., a type can be a child of another type in the entity type hierarchy. As embedding-based approaches emerged, more contextual information was considered in the entity type classification task (Section 5.2.1). Combining local, i.e., mention-level, and global, i.e., corpus-level, decisions was shown to be effective in improving performance (Yaghoobzadeh and Schutze, 2015).

We highlight one type of future work related to entity typing: *dealing with dynamic type systems*, which are type systems that have to be updated over time. Here it is realistic to expect that new types will be introduced over time. Dalvi *et al.* (2016) address the challenge of discovering new entity types with exploratory learning, which allows for classification of datapoints to a new type not found in the training data. Their approach can be improved by learning the association between a newly discovered type and existing types. Another interesting direction

is automatically labeling each new type. Along this line of work, Hovy (2014) considers an unsupervised approach to learning interpretable, domain-specific entity types from unlabeled text. He assumes that any common noun in a domain can function as potential entity type, and uses those nouns as hidden variables in a Hidden Markov Model to learn entity types in a new domain.

5.3 Entity-centric document filtering

Document filtering has been a traditional task in TREC in the form of Topic Detection and Tracking (TDT). TDT addresses event-based organization of broadcast news. The goal of TDT is to break an incoming text down into individual news stories, to monitor the stories for events that have not been seen before and to gather stories into groups that each discuss a single news topic (Allan, 2002). In contrast with ad-hoc search, where the collection is static, in TDT the queries are static while the document collection is dynamic and continuously updated in a streaming fashion.

The two main shared tasks on the completion and maintenance of KGs are Knowledge Base Acceleration (KBA) and Knowledge Base Population (KBP). Document filtering belongs to the KBA paradigm, in which the maintenance of KGs is performed by periodically recommending a number of relevant documents to KG editors, who will decide whether a document actually contains new facts and formulate the specific inclusion of facts in the KG. KBP, in contrast, aims to automatically populate a KG without the involvement of human editors, typically by applying relation extraction techniques to a document collection. A typical step in KBP involves the selection or filtering of documents from which specific relations are to be extracted, since applying relation extraction on all documents can be computationally expensive. Hence, KBA and KBP are complementary paradigms aimed at the construction of KGs.

Entity-centric document filtering is the task of analyzing an ordered stream of documents and selecting those that are relevant to a specific set of entities. Introduced at the TREC KBA track (Frank *et al.*, 2012), various approaches have been proposed to tackle this problem.

Definition 5.3. Given a stream of documents $\hat{D}$ and an entity e, the *entity-centric document filtering* task is to decide whether a document $d \in \hat{D}$ contains important information about e.

5.3.1 Approaches to entity-centric document filtering

Approaches to entity-centric document filtering can be grouped into two types: *entity-dependent* and *entity-independent*. We group them as such as they are the main distinguishing patterns of each method. Entity-dependent approaches learn a single model for each entity and aim to detect particular features of specific entities in a stream. In contrast, entity-independent approaches do not learn the specifics of each entity directly, but rather compare the distributional features of the entity against incoming documents to decide the relevance.

As this area is quite active, we cover the different approaches that emerged over the years.

Entity-dependent approaches to entity-centric document filtering

When TREC KBA was first held in 2012, most methods used by participants relied on *entity-dependent*, highly-supervised approaches utilizing related entities and bag-of-word features (Frank *et al.*, 2014). Here, the training data is used to identify keywords and related entities, and classify the documents in the test data.

Liu *et al.* (2013) present a typical entity-dependent approach. They pool related entities from the profile page of a target entity and estimate the weight of each related entity with respect to the query entity. They then apply the weighted related entities to estimate confidence scores of streaming documents. This approach achieves an F_1 score of 0.277 on the TREC KBA 2013 dataset; the official name-matching baseline obtained an F_1 score of 0.290 that year.

Dietz and Dalton (2013) propose a query expansion-based approach on relevant entities from the KG. They augment the original query terms (i.e., entity name) with other terms that are likely to indicate relevant documents, thus building a representation of the entity in the process. The way they approach this as a retrieval task, they cannot address the novelty aspects of the task, and evaluate a memory-less

method where predictions are not influenced by predictions on previous time intervals. Efron *et al.* (2014) also use an approach based on queries. They propose to find "sufficient queries," that is, high-quality boolean queries that can be deterministically applied during filtering. With this approach, no scoring is necessary since retrieval of entity-centric documents is purely based on these boolean queries. On the TREC KBA 2013 dataset, this sufficient query approach achieves an F_1 score of 0.316.

Representing entities through latent classes, Wang *et al.* (2015a) propose a discriminative mixture model based on introducing a latent entity class layer to model the correlations between entities and latent entity classes. This latent entity class is inferred based on information from a Wikipedia profile and category. They achieve increased performance by inferring latent classes of entities and learning the appropriate feature weights for each latent class. Since the model includes latent classes, the parameters of the model are learned with an expectation maximization (EM) procedure. In addition to entity features, their approach also takes into account hidden class features. This approach achieves state-of-the-art performance on the TREC KBA 2013 dataset, obtaining an F_1 score of 0.407.

Entity-independent approaches to entity-centric document filtering

Models that rely less on the specifics of each entity began to emerge during later years of the TREC KBA campaign (Balog, 2018). Balog *et al.* (2013) propose one such *entity-independent* approach. They study two multi-step classification methods for the stream filtering task, contrasting two and three binary classification steps. Their models start with an entity identification component based on alternative names from Wikipedia. They introduce a set of features that have become commonly used in subsequent TREC KBA campaigns. Evaluated on the TREC KBA 2012 dataset, this entity-independent approach achieves an F_1 score of 0.360 which is on par with the best performing methods of that year. To gain more insights, Balog and Ramampiaro (2013) perform an experimental comparison of classification and ranking approaches for this task. Their main finding is that ranking outperforms classification

on all evaluation settings and metrics on the TREC KBA 2012 dataset.

Similarly, Bonnefoy *et al.* (2013) introduce weakly-supervised, entity-independent detection of the central documents in a stream. Zhou and Chang (2013) study the problem of learning entity-centric document filtering based on a small number of training entities. They are particularly interested in the challenge of transferring keyword importance from training entities to entities in the test set. They propose novel meta-features to map keywords from different entities and contrast two different models: linear mapping and boost mapping.

Wang *et al.* (2013a) adopt the features introduced in (Balog and Ramampiaro, 2013) and consider additional citation-based features, experimenting with different classification and ranking-based models. They achieve the best official performance for document filtering in TREC KBA 2013 with a classification-based approach, obtaining an F_1 score of 0.330.

In contrast to earlier years, TREC KBA 2014 focused on long-tail entities, and less than half of the entities in the test set for that year have a Wikipedia profile (Frank *et al.*, 2014). In 2014, Jiang and Lin (2014) achieved the best performance (F_1 of 0.533) using an entity-dependent approach that uses time range, temporal, profession, and action pattern features in combination with classification-based filtering. Another notable approach that year summarizes all information known about an entity so far in a low-dimensional embedding (Cano *et al.*, 2014).

5.3.2 Relation of entity-centric document filtering to other tasks

Document filtering is related to other tasks mentioned in this chapter, in particular *entity recognition* (see Section 3.2) and *relation extraction*, which we will discuss in the next section (Section 5.4). It also has a connection with *entity discovery* (Section 5.1).

Document filtering uses named entity recognition to extract entity features from the candidate document. Having a good entity recognition component is important to allow us to extract useful features for filtering. Running relation extraction systems on a collection with a large number of documents can be very expensive computationally, which makes it

difficult to apply on a web scale. Document filtering methods can be used to select a pool of documents for performing relation extraction on a large scale. Finally, document filtering can be used to help build an initial profile for entity discovery by selecting relevant documents in which the entity appears.

5.3.3 Outlook on entity-centric document filtering

Looking back, there are two general strategies for document filtering. Initial approaches tended to focus more on an *entity-dependent* strategy, utilizing text classification or language modeling techniques (Section 5.3.1). In the text classification approach, a model will be learned for each entity based on training labels. Alternatively, language modeling may be used to rank documents based on the relevance/similarity to an input entity. More recent approaches consider more information, including the prior knowledge embedded in a document that will be filtered. The idea is that the type of entities will have relationships with the document type, and also that different filtering strategies will need to be applied depending on the document type.

With the *entity-independent* strategy, generic features extracted from the entity and document pairs are used to perform the filtering (Section 5.3.1). This strategy has the benefit of being applicable to perform filtering on entities not seen in the original training data. It was shown that context similarity and temporal information such as mention burstiness features are important features for this strategy (Balog *et al.*, 2013). Once the features have been extracted, the filtering step is typically formulated as a classification or learning-to-rank problem. In the classification model, the goal is to predict whether an entity is central in the document or not. In the learning-to-rank model, the goals is to rank documents based on their relevance to the input entity.

Two future directions can be identified for entity-centric document filtering: improving the filtering performance on *long tail entities*, and designing filtering approaches that can be applied to *unseen entities*. Signals for filtering entities can be different for head and tail entities. And having a filtering model that can be applied to unseen entities would be important in a practical setting. Reinanda *et al.* (2016) intro-

duce a document filtering approach focused on long-tail entities (see Section 5.3). For document filtering, features can be extracted from a target document, i.e., *intrinsic features*, or from "extrinsic" information around the entities, e.g., mention bursts. Reinanda *et al.* introduce several intrinsic features that can be extracted only from the documents and learn a single, global model for entity-centric document filtering that can be applied to long tail entities and entities not found in the training data.

5.4 Relation extraction and link prediction

Relation extraction originated in the original slot filling information extraction tasks that were first introduced in the Message Understanding Conference (MUC) series. The Automatic Content Extraction (ACE) evaluation campaigns (Doddington *et al.*, 2004) formally defined and included the task in 2002. To build a KG from entity relations from scratch, relationships between entities must be extracted, a task that is commonly known as relation extraction in the natural language processing community.

Definition 5.4 (Relation extraction)**.** Given a sentence s containing a pair of entities e_1 and e_2, decide whether e_1 and e_2 are connected through a relation of type r.

Incompleteness of KGs drives a lot of research in the area of knowledge base completion, in particular link prediction. Link prediction can be formally defined as follows:

Definition 5.5 (Link prediction)**.** Given a set of facts F where each fact is a triple of entity relations in KG, predict the existence of other relations between two entities e_i and e_j within relation type r, where e_i, e_j is in KG.

5.4.1 Approaches to relation extraction and link prediction

We briefly discuss approaches to relation extraction and continue with more detailed discussions on link prediction below. We refer to a survey on relation extraction by Bach and Badaskar (2007) for a more

comprehensive introduction to relation extraction. Our discussion of link prediction methods is partially inspired by Nickel *et al.* (2016). We classify approaches to relation extraction as follows: *supervised*, *distant supervision*, and *semi-supervised pattern extraction. Supervised approaches* rely on having training labels for each relation instance and contextual text expression the relations and then apply either features or learns kernel functions to classify the training data correctly. *Distant supervision* approaches utilize known relations, but without having the context in which the relations are expressed in a piece of text. Therefore, any extracted context will be considered and will contribute in the prediction. *Semi-supervised approches* rely on extracting textual patterns around known relations and using them to discover more relations.

Supervised, feature-based approaches to relation extraction

Supervised methods for relation extraction are typically grouped into two classes: *feature-based* and *kernel-based* methods. In the feature-based methods, syntactic and semantic features are extracted from the text. Syntactic features often include the entities, the types of the entities, word sequences between the entities, and the number of words between the entities. Semantic features are derived from the path in the parse tree containing the two entities (Kambhatla, 2004; Zhao and Grishman, 2005; GuoDong *et al.*, 2005).

To take advantage of information such as parse trees and to avoid generating features explicitly, kernel methods are introduced. Examples are presented in their original representation, and a function within the machine learning algorithm will compute the similarity between training examples within this rich representation. This representation can be in the form of a shallow parse tree or a dependency tree (Bunescu and Mooney, 2005; Culotta and Sorensen, 2004; Zelenko *et al.*, 2002).

Distantly-supervised, feature-based approaches to relation extraction

Another way of dealing with generating training data for distant supervision is based on pseudo-training data. Mintz *et al.* (2009) pioneered the work in this area. Later on, Riedel *et al.* (2010), Yao *et al.* (2010),

Hoffmann *et al.* (2011), Surdeanu *et al.* (2012), and Alfonseca *et al.* (2012) further refine the model by relaxing the assumptions introduced in the original method. For example, Surdeanu *et al.* (2012) achieve this by assuming that at least one distant-supervision training example is correctly labeled.

More recently, Zeng *et al.* (2015) propose a distant supervision model that takes into account the uncertainty of instance labels during training. Their model also automatically learns relevant features, avoiding the necessity of feature engineering. They do so by adopting a convolutional neural network architecture with piecewise max pooling. Semantic features include the path between the two entities in the dependency parse.

Semi-supervised pattern extraction

Since labeled data is expensive to create at scale, some work has started to investigate bootstrapping/semi-supervised approaches (Brin, 1998; Agichtein and Gravano, 2000). The main idea is to start with a small number of seed relation instances, learn a general textual pattern that will apply to these relations, and apply the newly discovered patterns to discover more relations. Later, web scale approaches for pattern extraxtion are introduced by Etzioni *et al.* (2004).

Work on semi-supervised pattern extraction further evolved into *open relation extraction* (Open IE). In work by Banko *et al.* (2007), relations are extracted without normalizing them to a specific schema. Relation-like tuples are extracted from text after learning how relations are typically expressed. Open relation extraction approaches are based on features such as the existence of verb and capitalizations of words. Wu and Weld (2010) leverage the alignment between Wikipedia infobox attributes and the corresponding sentences to automatically generate training data.

Open IE systems are prone to two challenges: *incoherent extraction* (cases where an extracted relation phrase has no meaningful interpretation) and *uninformative extraction* (where extractions omit critical information). Etzioni *et al.* (2011) address these challenges by developing a model on how relations and arguments are expressed in

English-language sentences, introducing a set of generic syntactic and lexical constraints in their system.

Approaches to link prediction are either based on modeling *latent features* or *existing connections in graphs*. *Latent-feature approaches* model the attributes of entities (including relationships to other entities) to learn latent representations of entities that can be used to predict links between two entities. In contrast, *graph-based approaches* apply graph algorithms (e.g., random walks) to discover potential connections between entities, and then compute the likelihood of the relation.

Latent feature models to link prediction

Factorization models learn a distributed representation for each entity and each relation, and make predictions by taking the inner products of the representations. Sutskever *et al.* (2009) are the first to propose the latent factor model approach to learning entity representations. Their approach utilizes learning a lower-dimensional representation of an entity while taking into account relation types by applying Bayesian clustering factorization techniques. The distributed representation is learned for each argument of the relation.

One of the simplest latent feature models is the *bilinear model*. Nickel *et al.* (2011) and Nickel *et al.* (2012) present RESCAL, which predicts triple likelihood through pairwise interactions of latent features. RESCAL works by modeling the likelihood score of a triple (a, r, b) as a bilinear model that captures the interaction between two entity vectors using a multiplicative term. During training, both the latent representation of entities and how they interact are learned. The method introduced in (Jenatton *et al.*, 2012) also belongs to this category, focusing on addressing the challenge of multi-relational data, in which multiple relations between entities may exist. This model also has a bilinear structure.

Socher *et al.* (2013) introduce an expressive neural tensor network suitable for reasoning about relationships between two entities. Although most of the work in this area represents entities as discrete atomic units or with a single entity vector representation, they show that

performance can be improved when entities are represented as an average of their constituting word vectors. In addition, they also show that these entity vectors can be improved when initialized with vectors learned from unsupervised large corpora. Their model can classify unseen relationships, extended from previous work (Socher *et al.*, 2012). Here, each relation triple is described by a neural network and pairs of entities are given as input to the relation's model. The neural tensor network replaces a standard linear neural network layer with a bilinear tensor layer that directly relates the two entity vectors across multiple dimensions. The model computes a score of how likely it is that two entities are in a certain relationship.

Some methods aim to learn *structured embeddings*, i.e., variants of latent distance models. Bordes *et al.* (2011) propose a model that learns to represent elements of any knowledge base into a low-dimensional vector space. The embeddings are established by a neural network where the architecture allows one to integrate the original data structure within the learned representations. The model learns one embedding for each entity and one operator for each relation. After the embeddings have been learned, kernel density estimation can be applied to estimate the probability density within that space so that the likelihood of a relation between entities can be quantified. This approach also combines a multi layer perceptron with bilinear models.

Also within the structured embedding paradigm, Bordes *et al.* (2014) propose a semantic matching energy (SME) function that relies on a distributed representation of multi-relation data. A semantic energy function is optimized to be lower for training examples than for other possible combinations of symbols. Instead of representing a relation type by a matrix, it is represented by a vector that shares the status and number of parameters with entities.

The following *translation models* are build on structured embeddings. The main feature of translation models is that the mapping between two entities is obtained by applying a relation vector, instead of matrix multiplication. Bordes *et al.* (2013) propose TransE, a method that models relationships by interpreting them as translations operating on the low-dimensional embeddings of entities. For two entities e and e', the embedding of entity e should be close to the embedding of

entity e' plus some vector that depends on the relationship between the two entities. It learns only one low-dimensional vector for each entity and each relationship. The main motivation is that translations are the natural way of representing hierarchical relationships that are commonly found in knowledge bases.

Wang *et al.* (2014b) extend the work of Bordes *et al.* (2013) and propose TransH, an improvement of TransE that considers certain mapping properties of relations including *reflexive*, *one-to-many*, *many-to-one*, and *many-to-many* relations. In addition, they propose an improved method to sample negative examples for the purpose of reducing false negative labels in training.

While TransE and TransH put both entities and relations within the same semantic space, an entity may have multiple aspects and various relations may focus on different aspects of entities. Generalizing the translation models even further, Lin *et al.* (2015) propose TransR, a method to build entity and relation embeddings in separate spaces and then build translations between projected entities. Ji *et al.* (2015) also propose an extension of TransR. They define two vectors for each entity and relation. The first vector represents the meaning of an entity or a relation. The other vector represents a way to project an entity embedding into a relation vector spaces. This means that every entity-relation pair has a unique mapping matrix. Yang *et al.* (2015) show that existing models such as TransE and TransH can be generalized as learning entities as low-dimensional vectors, and relations as bilinear/linear mapping functions for these entities. In terms of performance, DistMul (Yang *et al.*, 2015) is the current state-of-the-art approach for learning entity embeddings in the context of link prediction. For this task, the performance of different methods is often assessed using two common datasets, WordNet (WN) and Freebase (FB15K) following the evaluation setup introduced in (Bordes *et al.*, 2013).

Luo *et al.* (2015b) also consider the problem of embedding KGs into continuous vector spaces, but they consider another important dimension: *indirect relationships*. Existing embedding methods can only deal with explicit relationships within each triple (i.e., local connectivity patterns) while ignoring implicit relationships across different triples (i.e., indirect relationships through an intermediate node). Luo *et al.* present a

context-dependent KG embedding method that takes into account both types of connectivity pattern and obtains more accurate embeddings when used as representations for link prediction and triple correctness classification. The contextual connectivity patterns are learned through a contextual word embedding model such as skip-gram and continuous bag-of-words (CBOW) (Mikolov *et al.*, 2013). When applied on top of existing embedding methods such as structured embedding (SE) (Bordes *et al.*, 2011), SME (Bordes *et al.*, 2014) and TransE (Bordes *et al.*, 2013), it achieves an improvement over a random context baseline.

Graph-based models for link prediction

Rather than learning features of all entities and their pairwise interactions, graph-based random walk models utilize observed features found in existing connections. One such model is the path ranking algorithm (PRA) (Lao and Cohen, 2010) that performs random walks of bounded lengths to predict relations. PRA learns the likelihood of each relation path, combining a bounded number of adjacent relations.

One advantage of random walk models compared to latent factor models is their computational simplicity, although they tend to have lower inference accuracy due to the sparsity of connections in the graph. Gardner *et al.* (2013) aim to improve the effectiveness of PRA by enriching KGs with additional edges. These additional edges are labeled with latent features mined from a large dependency-parsed corpus of 500 million web documents. This enrichment is important to successfully improve the performance of PRA on a test set built from NELL knowledge base relation instances. Kotnis *et al.* (2015) propose a method for knowledge base completion using bridging entities. Previous work has enriched the graph with edges mined from a large text corpus while keeping the entities (i.e., nodes) fixed. Kotnis *et al.* (2015) augment a KG not only with edges but also with so-called bridging entities mined from web text corpus that allow the inference of missing relation instances. PRA is then applied to perform inference over this augmented graph, which helps to discover more relations.

Another approach to improving the performance of random walk models by addressing sparsity is introduced by Liu *et al.* (2016). They

propose a hierarchical inference algorithm that addresses the main problem of random walk models. They assume that entity relations are semantically bidirectional and exploit the topology of relation-specific subgraphs. From these assumptions, Liu *et al.* (2016) design a model that combines global inference on an undirected knowledge graph with local inference on relation-specific subgraphs.

Gardner and Mitchell (2015) extend PRA in a different way than is done in (Gardner *et al.*, 2013; Kotnis *et al.*, 2015; Liu *et al.*, 2016). They propose a simpler random walk algorithm that generates feature matrices from subgraphs. This method is proven to be more expressive, allowing for much richer features than paths between two nodes in a graph.

Yet another random walk model that uses observed features is proposed in (Toutanova and Chen, 2015). They show that the observed features model is most effective at capturing information present for entity pairs with textual relations. Another important finding is that a combination of latent and observed feature models will give the best performance. They incorporate both observed features from the KG and also textual evidence, similar to (Riedel *et al.*, 2013). Toutanova *et al.* (2015) propose a model that captures the compositional structure of textual relations and jointly optimizes entity, knowledge base, and textual relation representations. The proposed model significantly improves over a model that does not share parameters among textual relations with common sub-structure, achieving a ∼8% improvement in terms of mean reciprocal rank (MRR) on the FB15K-237 dataset.

5.4.2 Relation of relation extraction and link prediction to other tasks

Relation extraction uses entity recognition (see Section 3.2) for candidate selection and feature extraction; the documents from which the relations will be extracted are annotated with *entity recognition*. Relation extraction typically will only run on a selected pool of documents because applying it on a whole document collection would be too costly. Therefore, it would make sense to apply a *document filtering* (Section 5.3) component to run on the initial corpus. The output of relation

extraction needs to be estimated by *quality estimation* components, which we will discuss in Section 5.5.

Entity relations extracted by relation extraction and link prediction components can be leveraged by *entity linking* (Section 3.1) as an additional context for disambiguation. For *document retrieval* (Section 4.1), knowing the relationships between entities can be used to enrich document representations, for example through related entities. For *entity retrieval* (Section 4.2), relations can be used to enhance the entity representations. Finally, relationships are important as a basis for *entity recommendation* (Section 4.3).

5.4.3 Outlook on relation extraction and link prediction

Research around relation extraction started with supervised text classification-based approaches at the sentence level. Given sentences containing entities and the relationships between them, syntactic and semantic features are extracted to classify the relationship. Later variations of this approach avoided generating features explicitly but instead relied on various kernel methods to learn any entity relationships patterns. Due to the high degree of lexical variations in expressing entity relationships, preparing training data at the sentence level was deemed to be too costly. Meanwhile, *semi-supervised pattern extraction* methods focused on learning generic textual patterns pertaining to entity relationships. After that, the discovered entity pairs were used to learn more patterns (Section 5.4.1).

Distant supervision approaches aimed to alleviate the challenge of obtaining training data by relying on aligning known entity relationships to text. In the initial distant supervision strategy, evidence from multiple sentences that contain pairs of entities are combined to perform relation extraction for those pairs (Section 5.4.1). Later approaches then emerged to refine these initial approaches. The refinements mostly revolved around two ideas. The first was to relax the assumptions used in the training of the distant supervision model, correcting any noise that might be generated as a part of the approach. There are two general challenges to be addressed here: (1) the fact that text alignment might be incorrect, i.e., not all sentences containing a pair of entities express

the aligned relationships, and (2) the fact that a pair of entities might be connected by more than one relationship. Variations of distant supervision approaches were developed to address these challenges and one particular approach jointly considered these two challenges in the same model (Surdeanu *et al.*, 2012). Later iterations also started to consider the uncertainty of instance labeling explicitly in the model. The second, orthogonal refinement strategy combines distant supervision with neural methods to do away with feature extraction altogether.

Latent factor approaches to link prediction rely on learning the representations of entities and relationships. One of the first of these aimed to learn different representations depending on an entity as well as its role in the relationships, causing its representation as a subject or an object to be different (Sutskever *et al.*, 2009). Later methods then evolved where the representations do not rely on the occurrence of the subject or objects anymore. Inspired by tensor factorization, bilinear models that capture the relationships between two entity vectors using multiplicative terms were considered. The key idea was *collective learning*, i.e., all direct and indirect relationships should have a determining influence in the learned representation. In the next iteration of this approach, the factorization model started to consider the fact that multiple relations may exist between two entities. This line of work is then refined further: instead of considering entities as discrete atomic units, they are represented as the average of the constituting word vector of the mentions. This has the benefit of making the approach applicable to predict links on unseen entities. This latest approach in particular combined bilinear models of tensor factorization with a neural network (Socher *et al.*, 2013).

Structured embedding approaches are based on latent distance models; here, the probability of a relationship is estimated from the distance between latent representations in order to perform link prediction. Later iterations built upon this idea by refining the entity and relationship representation strategies. This started with a model that performs matrix multiplication on entity representations in a relation matrix, and then simplified into translation models which replace the relation matrix with a translation vector that can be applied to entity representations (Section 5.4.1). Certain refinements to the models were introduced,

explicitly catering for cases where relations can be reflexive, one-to-one, etc.

Up to this point, entities and relationships were considered to be in the same "semantic" space. This was refined by treating the translation between entities and relationships in relation-specific entity spaces, aimed to capture the fact that relations might involve different *aspects* of entities. Later on, this was refined even more by taking implicit relationships across different intermediate nodes into account.

With the latent factor and structured embedding approaches, we have seen two ways of representing entity relations: either as bilinear models or as translation vectors. Despite this difference, most of these approaches were generalized as learning entity vectors with a neural network, with the relations as bilinear and/or linear mapping functions for these entities (Yang *et al.*, 2015).

Finally, *graph-based* approaches to link prediction can be considered as methods that spawned from path ranking algorithms. The key idea here is to score entity pairs by a linear function of the features obtained from the paths between entities. Initial work in this area gave rise to two families of improvements. First, as graphs can be sparse, some methods were designed to address this sparsity. This is typically achieved by adding more entities and links to the graph, e.g., mined from a web corpus, and then performing the path ranking inference on the enriched graph. Alternatively, inference on a global graph is combined with a more local form of inference on subgraphs that are extracted for specific relationships. The second family focused on improving efficiency. As the original path ranking algorithm inference relies on performing expensive random walk probability computations, it was shown that optimizing this process through a sampling procedure can help improve the efficiency while maintaining effectiveness (Gardner and Mitchell, 2015).

We conclude that latent factor, structured embedding, and graph-based approaches rely on different intuitions and that a combination of graph-based and latent factor approaches have demonstrated to be very effective for link prediction (Toutanova and Chen, 2015).

Interesting research directions on relation extraction and link prediction include the following: *leveraging multiple sources for knowledge*

base completion and *targeted knowledge base completion with a budget*. We briefly discuss these directions below.

With the increasing availability of *heterogeneous resources*, it would be interesting to explore them for KG completion. Zhong *et al.* (2015) study the problem of jointly embedding a knowledge graph and a text corpus. The key issue is the alignment model that makes sure that the vector of entities, relations, and words are in the same space. They propose an alignment model based on the text description of entities. A possible extension of this work would be to incorporate semi-structured data (e.g., extracted open relations) in combination with existing facts and text data for knowledge base completion.

In the case where performing extraction on all documents in a collection is not feasible, extraction can be limited to specific entities and relation types of interest. Targeted knowledge base completion aims to discover new facts on specific relation types or entities. West *et al.* (2014) propose utilizing a question-answering inspired approach for performing targeted relation completion. Related to this, Hegde (2015) addresses the challenge of KG sparsity by focusing the completion on a set of target entities only. An interesting extension along this line of work is a targeted knowledge base completion system with a budget, i.e., a system that can automatically make the decision which entities or relations should be targeted first for extraction given limited resources or a limited budget. Adapting techniques from reinforcement learning would probably be suitable in this setting.

5.5 KG quality estimation

Automatic quality estimation of KGs is a relatively new area. Little work has been devoted towards ensuring the quality of facts contained in knowledge graphs. Work on this field can be divided into two main areas: evaluating the overall quality of a set of facts in a KG, and evaluating the quality of each individual unit in the KG, i.e., at the triples or edit level. Within each area, there are various paradigms and approaches that we discuss below.

5.5.1 Approaches to KG quality estimation

We start the discussion by discussing quality estimation at the set level, and continue to discuss various paradigms at the unit level.

The *KG correctness* setup focuses on estimating the overall quality of a set of facts in the KG. Most of the existing focus is on *correctness*, i.e., on how closely the facts in the KG reflect the real world.

Sampling-based approaches to KG correctness estimation

One way to estimate the correctness of the facts covered in a KG can be by sampling a number of facts and evaluating them manually. Simple random methods are often not efficient to estimate the true accuracy of a KG as the annotions can be costly. Therefore, the challenge would involve selecting a set of triples from the KG that would allow us to get a good estimate of the correctness. Sampling-based approaches to triple correctness prediction focus on discovering the right sampling strategies that would minimize the number of annotations required while obtaining a reliable estimate.

Ojha and Talukdar (2017) introduce KGEval, a method which rely on *coupling constraints*, i.e., the idea that some facts in the KG are connected and, for each group of connected facts, evaluating a representative subset of the group facts would be sufficient. These *coupling constraints* can be derived from the ontology of the KG and also link prediction algorithms. Relationships between facts are established trough the coupling constraints and stored in the form of a graph. When a fact is evaluated manually, the correctness of related facts is then inferred from the graph. This process is performed iteratively until the estimate of the correctness of the full KG converges.

Similar to (Ojha and Talukdar, 2017), in (Gao *et al.*, 2019) another method is introduced to efficiently sample a number of triples from the KG. Instead of focusing on the relationship beween facts, Ojha and Talukdar (2017) focus on computing the minimum number of annotations required to get a reasonable correctness estimate. They experiment with various sampling techniques. The most efficient technique, *two-stage weighted cluster sampling* involves grouping triples by the subject entities. First the entities that are going be evaluated are sampled, then

a subset of triples from the selected entities are sampled from each cluster. Furthermore, they also introduce an extension of the method that could work on evolving KG, allowing incremental evaluation of the correctness to performed. Compared to the method introduced in (Ojha and Talukdar, 2017), the two-stage cluster sampling method is more efficient, as it does not require expensive inference operations to be performed.

Next, we discuss paradigms for evaluating the quality of KGs at the unit level. The *triple correctness prediction* paradigm focuses on estimating the correctness of a knowledge base relation triple by considering the sources of the fact and also the confidence around the extraction method. The *contribution quality estimation* paradigm focuses on the estimating the quality of each changes to the KG made as individual contributions by looking at various features around the update. Another paradigm, *vandalism detection* focuses on the detection of malicious edits on publicly avaiable KGs to detect anomalous updates and estimate the quality.

The *triple correctness prediction* paradigm focuses on estimating the correctness of a single KG relation triple.

Fusion-based approaches to triple correctness prediction

Dong *et al.* (2014b) estimate the probabilities of fact correctness from multiple sources. They combine confidence scores from several text-based extractors and prior knowledge estimated based on known facts from existing knowledge repositories. These scores are then fused and converted into a probability with a technique called Platt scaling. The approach utilizes multiple relation extractors based on the text, html structure, and microformat annotations on the web. Dong *et al.* fuse the output of this system with a graph-based prior inferred from the current state of the knowledge graph. They consider two methods to compute graph-based priors: (1) a path ranking algorithm (PRA) (Lao and Cohen, 2010), and (2) an embedding method based on a multilayer perceptron.

The multilayer perceptron model is obtained by first performing a

low-rank decomposition of the KG represented as a tensor, obtaining the embeddings of triples in a lower dimensional representation. The output of the extractors and priors are then combined as features in a feature vector. Then, the weight of each feature is learned through a classifier such as linear logistic regression, or ensemble decision trees. To evaluate the performance of individual and combined approaches, Dong *et al.* (2014b) first generate a set of *confident facts*, i.e., facts that have an estimated probability of being true above 90%. Then, they sample a balanced number of triples per relation type from this set, and compare the triples against the triples in Freebase.

Building on (Dong *et al.*, 2014b), Dong *et al.* (2014a) compare different methods of aggregating knowledge, inspired by data fusion approaches. An approach to finding systematic errors during data extraction is proposed in (Wang *et al.*, 2015b), while Dong *et al.* (2015b) propose a method to decompose errors made during the extraction process and factual errors in the web source. The extraction performance is evaluated by comparing the extracted triples against Freebase. Continuing this line of work, in (Li *et al.*, 2017) the knowledge fusion approach is further applied on long-tail verticals.

The *contribution quality estimation* paradigm focuses on estimating the quality of each individual contribution to a KG.

Feature-based approaches to contribution quality estimation

Tan *et al.* (2014) present a method for automatically predicting the quality of contributions submitted to a KG. The proposed method exploits a variety of signals, including the user's domain expertise and the historical accuracy rates of different types of facts; this enables the immediate verification of a contribution, significantly alleviating the need for post-submission human reviewing. The following signals are considered for prediction:

- **User contribution history** These features are meant to capture a user's reputation based on their previous contributions, such as the total number of prior contributions, total number of correct contributions, and fraction of correct contributions.

- **Triple features** These features are meant to capture the relative difficulty of each relation. This difficulty is estimated from the historical deletion rate of that particular predicate.
- **User contribution expertise** User expertise is estimated based on previous contributions. They consider three concept spaces: LDA topics, taxonomy, and triple predicates.

In their experiments, Tan *et al.* (2014) compute the *relative error reduction* (RER), defined as $\frac{error_{baseline} - error_{system}}{error_{baseline}}$, i.e., the reduction of the error of the system compared to the error of a baseline at the same recall level. Their approach achieves a relative error reduction (RER) of 60%, a substantial improvement over baselines based on the following strategies: majority, users' contribution history, and users' long term contribution quality.

Flekova *et al.* (2014) study the user-perceived quality of Wikipedia articles. They utilize a Wikipedia user feedback dataset that contains 36 million Wikipedia article ratings contributed by ordinary Wikipedia users. The ratings incorporate the following quality dimensions: *completeness*, *well-writtenness*, *trustworthiness*, and *objectiveness*. They select a subset of biographical articles and perform classification experiments to predict their quality ratings along each of the dimensions, exploring multiple linguistic, surface and network properties of the rated articles.

Graph-based approaches to contribution quality estimation

Li *et al.* (2015) consider the problem of automatically assessing Wikipedia article quality. They develop several models to rank articles by using the editing relations between articles and editors. They develop a basic quality model based on PageRank in which articles and editors are represented as nodes connected by edges that represent editing relations. Articles are ranked by node value. To take into account multiple editors, they incorporate contributions made to an article and utilize these as edge weights during the PageRank computation.

The *vandalism detection* paradigm focuses on detecting intentional vandalism actions on structured and semi-structured knowledge bases.

Vandalism is defined as malicious insertion, replacement, or deletion of articles.

Feature-based approaches to vandalism detection

Heindorf *et al.* (2015) introduce a corpus for vandalism detection on Wikidata and perform some initial analysis on vandalism on this particular corpus. Later, Heindorf *et al.* (2016) propose a set of features that exploit both content and context information. Content features include features at the character, word, sentence, and statement level. These include capitalization, character repetitions, profane/offensive words, and changes of suspicious lengths. Context features include the user, item, and revision features.

Research on vandalism detection originally considered unstructured and semi-structured knowledge bases, e.g., Wikipedia. Potthast *et al.* (2008) are the first to render vandalism detection as a machine learning task. They compile features for detecting vandalism on Wikipedia. Their method works at the *edit* level, where each edit consists of two consecutive revisions of a document. Each edit is then represented as a feature vector in which the classifier is applied. Currently, all vandalism detection approaches are *feature-based*.

5.5.2 Relation of KG quality estimation to other tasks

Quality estimation has direct connections to *relation extraction and link prediction* (Section 5.4). In these two tasks, we estimate the probability of the extracted triples being correct for the purpose of completing a KG. The correctness/quality of entity relations can be incorporated in *entity recommendation* systems (Section 4.3), to allow the weighting of recommendations based on the quality or validity of the relations.

Any approaches that use entity profiles, such as *document retrieval* (Section 4.1) or *entity retrieval* (Section 4.2), would benefit from the quality provided by the estimation methods, as the retrieved objects can be biased towards validated facts.

5.5.3 Outlook on KG quality estimation

Looking back, the quality of a KG can be estimated at the global level and at the individual triple level, and also by looking at the contribution quality and detecting vandalism. With a sampling strategy that considers the frequency of entities in the KG, the overall quality of a KG can be estimated efficiently. Furthermore, the quality can be estimated in an incremental fashion (Gao *et al.*, 2019).

When we want to estimate the correctness at the level of individual triples, it has been demonstrated that combining evidence from various text-based relation extractors, e.g., from text and tables, with graph-based and embedding-based methods to perform link prediction is effective (Section 5.5.1). Since scores from various methods might not have the same mean and variance and/or be calibrated properly, scaling/normalization will need to be applied. It was demonstrated recently that accurate calibration can also improve link prediction itself.

Contribution quality estimation and vandalism detection rely on various features derived from the text and revision history of a KG. Alternatively, the relationships between editors and articles that they contributed to have also been demonstrated as useful signals that can be utilized to estimate quality (Li *et al.*, 2015).

We expect quality estimation models that combine evidence from *multiple sources*, e.g., both text and graphs with more complex features, to appear in the future as such models can incorporate features extracted from articles or triples with relationship information between contributors or items.

Besides more complex models, *alternative validation strategies* (e.g., based on gamification or crowdsourcing) are an interesting research direction. One such strategy is presented in (Vannella *et al.*, 2014), where video games are used for validating and extending knowledge bases and this idea will most likely receive more attention in the future.

Finally, verification of the correctness of KG facts in long-tail verticals (i.e., in less popular domains), as explored in (Li *et al.*, 2017), still remains an interesting challenge.

5.6 Conclusion

In this chapter we have considered the contribution of IR techniques and methodologies to a range of KG-related tasks, including entity classification, entity discovery, entity typing, document filtering, relation extraction, link prediction, and quality control. A common theme is that IR helps KG-related tasks by providing techniques that can be used to identify entities and facts in the context of constructing, completing, and estimating the quality of KGs.

In the next chapter, Chapter 6, we follow up with an example of an end-to-end IR task in which KGs are put to work in practice: web search. There, we also describe an end-to-end pipeline for constructing a KG from scratch. After that, Chapter 7 continues with a discussion of the challenges encountered in tasks that we have discussed in Chapters 4 and 5; in Chapter 7 we also conclude the survey.

6

Applications

In this chapter, we describe two end-to-end pipelines that utilize information retrieval and knowledge graph techniques. The first focuses on web search and the second deals with building knowledge graphs from scratch using unstructured text in a document corpus.

6.1 Web search

We start with a discussion of an end-to-end web search pipeline utilizing knowledge graphs. As the individual techniques have already been discussed in detail in previous chapters we only provide a high-level overview here, point to relevant sections, and discuss additional work that does not fit within the tasks presented in the previous chapters.

Figure 6.1 shows an end-to-end pipeline of web search components leveraging KGs. Given a query, we discuss the KG-related steps that are executed in order to present a search engine result page (SERP). For the purpose of the discussion in this section, we assume that the goal is to build a SERP with a ranked list of documents, a direct answer, and a knowledge card (see Figure 6.2). We start with *query understanding*: identifying entities and intent-related terms in the query. The output of the analysis performed in this step will be used downstream

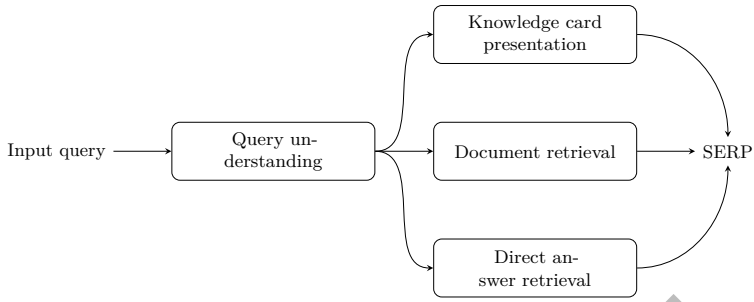

Figure 6.1: Web search pipeline with a KG.

during *document retrieval*, *direct answer retrieval*, and *knowledge card presentation*.

A similar pipeline can also be applied to mobile search as the main components are very similar. The chief distinction between mobile search and web search lies in (1) the screen presentation real estate that is available for presenting results, (2) the nature of interaction with users, and (3) the availability of a richer context in the form of, e.g., location-based information and interlinked apps. In mobile search, the presentation space is limited and therefore more work on finding and utilizing entity-related information under this limitation would be an interesting direction. Besides, mobile search tends to be more app-centric and personalized and pushing timely and relevant entity-related information to a specific user would make for an interesting strategy.

6.1.1 Query understanding

KGs are now being used to enrich query representations in an entity-aware fashion to encode for rich facts organized around entities. This type of enrichment can be performed by identifying entities mentioned in the query, and also identifying tasks associated with entities mentioned in the query.

Identifying entities mentioned in a query, i.e., entity linking, is an important part of query understanding. In the context of search, knowing that "london weather" refers to the weather in something called "London" — which may be disambiguated based on selecting the most common sense to the capital of the United Kingdom — would help in providing

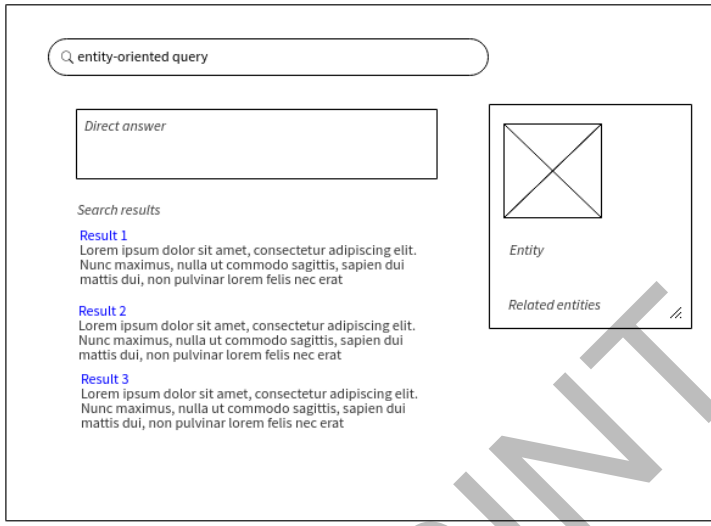

Figure 6.2: SERP with a ranked list of documents, a direct answer, and a knowledge card.

a better experience to the users. Pantel and Fuxman (2011) were one of the first researchers to publish a paper on entity linking in web search queries, estimating the relevance between a query string and an entity from query-click graphs. Blanco *et al.* (2015), on the other hand, learn entity representations using contextual information from Wikipedia and employ embeddings for disambiguation. Their approach can be extended by incorporating more contextual information from news, related queries, and trends. Learning useful signals from this contextual information is an important direction. We refer to Section 3.1 for more details on entity linking.

After understanding queries and identifying entities, understanding tasks around entities is the next step. Along this line of work, Pound *et al.* (2012) present an approach to compute structured representations of keyword queries over a reference KG. They mine common query structures from a web query log and map these structures into a reference KG. For example, the query "songs by the beatles" could be interested as a query about the entity "The Beatles," with the songs as the target entities that have relationship with the band. The goal of the approach

introduced in (Pound *et al.*, 2012) is to derive a structured keyword query that can be mapped to entities in a KG. Several authors have focused on recognizing intents and collecting them in a structured schema. Resolving query words to intents can be performed in a knowledge graph-based framework. For instance, Zhao and Zhang (2014) tackle the problem of query intent understanding by representing entity words, refiners, and clicked URLs as *intent topics* in a unified knowledge graph-based framework. Reinanda *et al.* (2015) focus on *entity aspects*: common search tasks around entities. They present an approach that builds a repository of common tasks for each entity from query logs and show the different ways this aspect can be leveraged to enhance entity-oriented result presentation and query recommendation. Examples of aspects for a person entity would be "net worth," "date of birth," "place of birth," etc. Dalton and Dietz (2013) propose constructing a so-called *knowledge sketch* that leverages KG data and relevant text documents to construct query-specific KG representation; here, a knowledge sketch is a distribution over entities, documents, and relationships between entities for a specific information need.

Another set of related work was created for the "actionable knowledge graph task" in the context of the NII Testbeds and Community for Information access Research (NTCIR) evaluation campaign (Jatowt and Yamamoto, 2017). Two main tasks made up this track. First, the *action mining subtask*, in which the goal is to generate a ranked list of actions given an entity and its entity type. Second, the *actionable knowledge graph generation subtask*, in which the goals is to rank entity attributes based on their relevance to the query. An example of action related to a query would be "visit a temple" for the query "kyoto budget travel". Here, the query might contain an entity, entity type, and also action. One goal for this evaluation campaign is to explore the construction of an Actionable Knowledge Graph (AKG), a KG that contains actions and their relationship to relevant entity or entity types.

More and more users are using web search engines to perform more complex tasks, such as creating a travel itinerary. These complex tasks are often associated with queries related to multiple entities of varying types. Wang *et al.* (2014a) consider the problem of understanding complex tasks and split it into three sub-problems: finding task-intrinsic

entities, generating a task name, and suggesting proper search results covering all desired entities for the complex task. They propose a model to automatically generate complex task names and related task-intrinsic entities (Wang *et al.*, 2014a). A complex task such as *travel to london for holiday* could spawn three tasks: reserving a ticket, booking a hotel, and researching for attractions.

6.1.2 Document retrieval

The next step after understanding the query, is retrieving web pages that are relevant to the query. Insights gleaned from the previous step could be utilized to improve the document retrieval process, e.g., if we could extract the likely intent for a task involving a certain entity type, we could consider these as features to be used during document retrieval. Here, we discuss how entities identified in a query can be leveraged for retrieval. In one of the earliest papers on leveraging entities in the form of Wikipedia articles, Gabrilovich and Markovitch (2007) propose Explicit Semantic Analysis (ESA), a method to represent the meaning of texts in a high-dimensional space of concepts derived from Wikipedia. They represent the meaning of any text as a weighted vector of Wikipedia-based entities. More recently, Xiong *et al.* (2017b) introduced a new ranking technique based on graph embeddings. First, queries and documents are represented in an entity space and they are subsequently ranked based on the semantic connections in their knowledge graph embeddings. We refer to Section 4.1 for a more detailed discussion on this step.

6.1.3 Direct answer retrieval

When applicable, sometimes we want to display direct answers and not just a listing of documents and snippets. Again, insights gleaned from the previous query understanding step could be utilized to decide whether a system should return direct answers. Should we consider that the user intent involves a single answer (e.g., a factoid query), we can further aim to provide a direct answer in addition to a ranked list of documents. Here, we discuss approaches for providing direct answers to user queries, supported by question answering techniques using KGs.

Queries in the form of natural language questions can be answered in different ways: from text, using KGs, or a combination thereof. Broadly speaking, two approaches for question answering exist, either based on *semantic parsing* or on *information extraction*. In semantic parsing, the question is first converted into a "meaning representation," and then this representation is converted into a structured query, which then will be passed on to the retrieval system. With the information extraction-based approach, a set of candidate passages is first obtained using retrieval techniques, then the answer is extracted from the candidate passages. Below, we discuss approaches to question answering from KGs.

Open-domain question answering, which returns exact answers to natural language questions issued by a user, can be viewed as a form of entity retrieval. Sun *et al.* (2015) consider the task of open domain question-answering by querying KGs. They propose an approach that mines answers directly from the web and employ KGs as an important auxiliary to further boost the question-answering performance.

Yao and Van Durme (2014) propose an automatic method for question answering from a structured data source. Their approach focuses on the information extraction angle to question answering, first performing information retrieval and later continuing with deep analysis on returned candidate answers. A retrieval-based approach is employed to retrieve a specific KGs topic node related to the question being asked. A subgraph is later expanded from the topic node and an analysis is performed on the subgraph to obtain the answer. Unlike typical knowledge-based question answering systems that transform natural language questions into meaning representations and perform answer retrieval using the generated meaning representation, Bao *et al.* (2014) present a translation-based approach to solving the two components in one unified framework.

Yih *et al.* (2015) propose a novel semantic parsing framework for question answering that utilizes a knowledge graph. They define a query graph that resembles subgraphs of the KG and can be directly mapped to a logical form. Semantic parsing is cast as query graph generation, commencing as a staged search problem. This query graph consists of four types of nodes: grounded entity, existential variable, lambda variable, and aggregation function. Grounded variables are

existing entities in the KG. Existential variables and lambda variables are ungrounded entities. Yahya *et al.* (2016) propose a method for querying and ranking on extended knowledge graphs that combine relational facts with textual web contents.

In (Berant *et al.*, 2013), SEMPRE, a method to train a semantic parser from question-answer pairs is presented. One of the biggest contributions at the time was allowing the learning of semantic parser to support predicates in different domains, and also learning this from question-answer pairs instead of annotated logical forms. Aligning phrases to predicates is a main component to solve this problem, however, phrases is a question might not be informative or even missing. To address this issue, they perform a bridging , i.e., generating predicates based on adjacent predicates rather than simply aligning them based on words.

In the database community there have been many efforts since the early 1960s on supporting natural language queries by translating them into structured queries; see e.g., (Green *et al.*, 1963; Woods, 1977; Bronnenberg *et al.*, 1980). For a comprehensive review on question answering approaches, we refer to (Bast *et al.*, 2016).

6.1.4 Knowledge card presentation

Another typical feature of a modern search engine are *knowledge cards*. Knowledge cards or entity cards that enhance users' search experience when searching for information related to entities. After entities have been identified in a query, the related information for that entity is obtained from a KG and displayed. Typically, entity-relation information such as key attributes and related entities are presented in a knowledge card. We refer to Section 4.3 for comprehensive discussions around recommending related entities to be displayed in a knowledge card. In this section, we briefly discuss some related work around knowledge cards.

For knowledge card presentations, Bota *et al.* (2016) study the effect of entity cards on search behavior and perceived workload. In (Shokouhi and Guo, 2015), proactive card recommendations that push relevant information based on user preferences are studied. Finally, search pages are becoming increasingly complex with the addition of entity cards

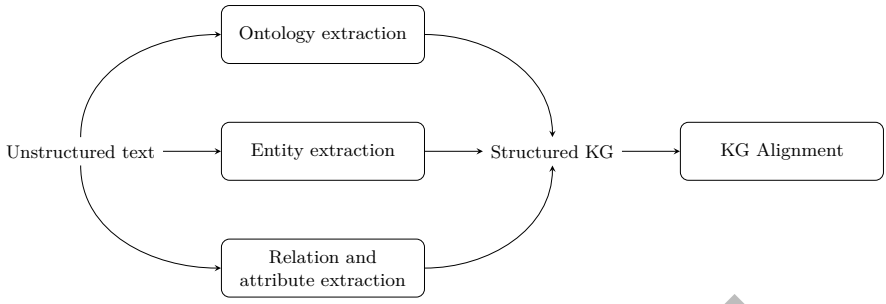

Figure 6.3: Knowledge graph construction from scratch.

and aggregated results; Navalpakkam *et al.* (2013) measure and model eye-mouse behavior in the presence of these nonlinear page layouts.

Mobile devices have limited visible display portions; optimizing presentation in such limited space in an important issue. Lagun *et al.* (2014) work on investigating a measurement of attention and satisfaction in mobile search based on the visible portions of a web page on mobile phones.

We have given a brief overview of an end-to-end web search pipeline utilizing knowledge graphs, starting with understanding an input query and ending with displaying entity-related information on a SERP.

Next, we examine another end-to-end pipeline that utilizes information retrieval and knowledge graph techniques, this time focusing on the knowledge graph aspects.

6.2 Knowledge graph construction

In this section, we discuss how to build a knowledge graph from scratch from unstructured text. Again, we mostly refer to sections in previous chapters and only discuss work that does not fit within the tasks already presented.

Figure 6.3 illustrates the end-to-end construction pipeline for KGs. A knowledge graph consists of entities, relations, and an ontology that describes the structure of the knowledge graph. Therefore, constructing a KG involves extracting or defining such entities, relations, and ontological constructs. In some cases, we might perform some of these steps manually depending on the availability of data. For example, we

might already have a set of entities of interest to be considered, e.g., from existing databases, which means that we do not need to extract the entities from scratch. However, if we are working in a new domain, we might need to extract and discover entities from text relevant to the domain, and then extract the relationships between those entities.

Finally, a new KG can also be aligned or combined with other KGs. This is particularly useful in the case where the knowledge graphs are complementary to each other, e.g,. when they are built from resources across different languages, but cover overlapping entities. This types of problem are also found in other domains (e.g., databases) and are related to tasks such as *ontology alignment*, *entity reconciliation*, *entity resolution*, and *entity blocking*.

6.2.1 Ontology extraction

When building a KG, one important ingredient is a type system for entities. An ontology describes a set of entity types and also how they are related. On a general purpose KG, the type system for people entities might include categories such as *politician*, *actor*, and *sporstman*. A type system could be hierarchical, i.e., forming a tree of classification of entities. An entity could be associated to multiple relevant types. This type system will be used for entity typing (Section 5.2). It can be defined manually or extracted from text, e.g., (Dalvi *et al.*, 2016; Hovy, 2014). In addition, the relations of interest that need to be extracted in the relation extraction phase (Section 5.4) can be defined manually or discovered automatically from text. As we discuss in Section 5.4, one way to discover relations from text is to extract potential relation triples using *open relation extraction* and normalizing them (Banko *et al.*, 2007).

6.2.2 Entity extraction

The first step of extracting entities from text involves detecting mentions of entities in the text. One way to achieve this is by applying named entity recognition techniques (see more in Section 3.2). Entities can have different mentions, e.g., the entity *Apple Inc* can be referred to as *Apple Inc*, *Apple Incorporated*, etc. Therefore, after mentions of entities have

been detected, different mentions referring to the same entity need to be linked. This can be achieved by applying cross-document coreference resolution methods, e.g., by grouping mentions of entities based on the similarity of their context.

Depending on the domain, not all entities mentioned in the text might be useful to be included as KGs entities. To build a set of entities of interest we need to decide the noteworthiness, i.e., whether the entity should be included in the KG. For example, if we extract the entities from text, we might not want to include all person entities that are extracted, but instead only the ones that might be popular and relevant for the KG being constructed. Entity discovery methods can be applied to discover such noteworthy entities (see Section 5.1).

Finally, an entity in a KG is typically equipped with a description or profile providing a high-level overview of the entity. To generate this profile, documents that are relevant to an entity can be selected through document filtering techniques (Section 5.3).

6.2.3 Relation and attribute extraction

At the early stages of relation and attribute extraction, we can discover entity relations with supervised relation extraction techniques, as we do not have any entity relations data at this stage and we need to build an initial set of facts. From a predefined set of relations, extractors are trained for each relation type. Mentions are detected within the unstructured text and linked to an entity node in the knowledge graph. The extractors can be trained in a supervised fashion, e.g., with binary extractors. Following this approach, we will have a set of relation extractors, where each is optimzed to detect a particular relations type, e.g., *place of birth extractor*, *nationality extractor*, etc. A set of extracted relations will form the initial facts of the newly-built KG.

Once an initial set of facts is available, more relations can be discovered with semi-supervised learning techniques such as distant supervision. Instead of extracting relations at the sentence level, distantly-supervised relation extractors typically build a corpus-level representation of candidate relations found in the text and try to predict the existence of relations from these possible pieces of evidence. Relations

known at this stage will be used as training data for which the distant supervision relation extractors will be trained against.

Finally, more relations can be discovered with KG-completion approaches, inferring new relations from currently known relations. For example, *nationality* relationship could probably be inferred from *place of birth* and other relations known in the KG. We refer to Section 5.4) for details on relation extraction and completion methods.

After the relations have been extracted, one important factor to consider would be correctness of the extracted triples. We refer to Section 5.5 for details on various quality estimation methods.

6.2.4 Knowledge graph alignment

Information that is available for improving KGs might come from different sources; hence it would be useful to combine KGs built from different sources—including different languages. The same entity might be referred to differently in different KGs, due to different identifier systems or different naming/mentions coverage. Furthermore, KGs that contain the same relationships might use different ways of labeling them, althogh the semantic of the relationship is the same. To achieve this, the entities in each KG must be aligned across KGs and languages. Aligning the relationships between entities from different schemas would be the next step; Galárraga *et al.* (2013) present an approach to perform this schema alignment based on rule mining. Assuming some of entities from two KGs have been aligned, rule mining techniques such as Amie (Galárraga *et al.*, 2013) can be applied to align the relationships. Using the aligned entities as reference points, we can discover relationships between properties and relationships in the KGs.

One automatic method to perform the alignment is to train multilingual embeddings. Chen *et al.* (2016) introduce a method to encode entities and relations of each language in a separate embedding space and provide translations for each embedding vector to its cross-lingual counterparts in other spaces. One particular feature of their approach is that the functionalities of each monolingual embedding are preserved. Wang *et al.* (2012) present an alternative, a linkage factor graph model to perform cross-lingual knowledge graph linking, i.e., linking an entity

in one KG to another. They formulate the problem as predicting the label of entity pairs between two KGs, using features from the citation graph (i.e., references providing evidence of attribuet or relationships) of the entities. Finally, Wang *et al.* (2013b) propose a transfer learning approach to complete Wikipedia infoboxes, i.e., extracting a particular attribute of a KG in the target language utilizing the information from a source KG. The task is formulated as a classification problem, where a missing infobox value in a target language is predicted from full text in the source language. Each word is represented as a feature vector built of format, POS tag, and token features.

Another approach to knowledge base alignment across languages relies on schema matching. Zhang *et al.* (2017b) present an attribute schema matching approach for KG completion across languages. They propose a model that leverages text, article, category and template features, and model the relationships between infobox attributes in cross-lingual KGs as factor graphs. A factor graph is a probabilistic graphical model that consists of variables and factors that describe relationships between the variables in the model. The parameters of the factor graphs are learned from known attribute mappings.

Next, we discuss work related to cross-modal KGs construction, i.e., combining different types of input data such as text, images, and video. In (Melo and Tandon, 2016), various methods of combining cross-modal KGs are summarized. The most popular among them, ImageNet, combines a broad WordNet schema with images representing objects; image-level and object-level annotations are obtained through crowdsourcing. Another cross-modal KG, Visual Genome is a KG centered around objects in images and relationships between the objects (Krishna *et al.*, 2016). It is also built using crowdsourcing and currently covers around 75,000 concepts, where the concepts are objects, attributes, and relationships between objects. Examples of concepts would be "a ball," "a person holding ball," etc.

In another publication, Zhu *et al.* (2015) present a method to build a large-scale multimodal KG for the purpose of answering visual queries. Their KG construction system involves three main steps: *data-preprocessing*, *factor graph generation*, and *high-performance learning*. First, raw data in the form of annotated images are pre-processed

into a structured representation, to allow efficient computing. Next, they write human-readable rules to define the KG's ontology. Their system automatically creates a factor graph from rules manually created to describe the relationship between the attributes in image annotations. Then, a scalable Gibbs sampler is applied to learn the weights in the factor graph. We refer to (Wu *et al.*, 2016a) for a comprehensive discussion on visual question answering, i.e., a task in which the goal is for systems to be able to answer questions about an image.

In this section, we have discussed how KGs can be built from scratch. We discussed techniques that start with defining or extracting the ontology, extracting entities, and extracting relations between the entities. Once KGs have been built, they can be aligned across languages and modalities with other KGs.

7

Conclusion and Discussion

7.1 Conclusion

The aim of this survey has been to give a broad overview of knowledge graphs (KGs) from an information retrieval (IR) perspective. Our overview has included methods that leverage KGs to improve IR as well as methods from IR that help improve KGs. While our emphasis has been firmly on matters related to IR, we have also included related work on language technology where we believed this would be beneficial.

Our strategy with this survey has been to provide a broad coverage of the interface of IR and KGs. Because of this, the amount of detail that we have provided is limited. In addition, we have emphasized recent work over old work. For areas that are broad enough to have their own survey, we have only covered key publications and have left a detailed elaboration of the area to an existing survey or tutorial. In this manner, we hope to have created a useful "hub" for exploring the exciting interface of IR and KGs.

The core of the survey is organized around two chapters, each of which starts with a task-oriented structured summary in which approaches for each task are grouped and contrasted. In addition, we have presented introductions to common tasks in a background chapter.

We have identified nine groups of tasks, four in the context of leveraging KGs for IR (Chapter 4), and five in terms of adapting IR techniques to improve KGs (Chapter 5). For each group of tasks, we have identified its origins, common techniques or families of methods used to address the task, the relation to other tasks, and an outlook. We provide algorithmic details and a general framework to work within the context of the task. We have also reflected on future developments related to the tasks.

To give context, we have presented applications of techniques and methods that we discussed in the core chapters. In Chapter 6 we have presented an end-to-end example web search pipeline utilizing KGs, starting with understanding an input query and ending with displaying entity-related information on a SERP. There, we have also discussed how KGs can be built from scratch, from discovering entities and relations to aligning KGs across languages and modalities. For both of these applications we have provided pointers to the specific techniques discussed in the earlier chapters and, where applicable, we discuss methods that do not fit naturally within the core chapters to provide complementary details.

As we said above, this survey is intended to give an overview of work related to KGs from an IR perspective. Beyond the direct uses of KGs in the context of IR as identified in Chapters 4 and 5, we believe that search-oriented conversational systems, result explainability, and domain-specific KG enrichment would make interesting applications. This survey would be useful for researchers and engineers who are interested in such applications.

7.2 Discussion

In the course of this survey, we have identified several common issues that are encountered across multiple tasks. We consider the following issues important when it comes to IR and KG-related tasks in particular: *ambiguity*, *sparsity*, *temporality*, *explainability*, and *user behavior*. Next, we briefly discuss each of these issues and conclude with general findings and an outlook.

We refer to *ambiguity* as uncertainty regarding semantic interpreta-

tions of the output resulting from an underlying component of machine learning models. Ambiguity is a very common problem when it comes to entity-oriented search or even when dealing with text generally. To give an example in the context of the entity linking task, there are two main kinds of ambiguity in entity linking: a single mention can refer to different entities and different mentions can point to the same entity. These two kinds of ambiguity affect the entity linking tasks the most. Since entity linking is an upstream task that may affect other tasks downstream, errors resulting from ambiguity in this step will propagate and thus will need to be avoided. Another example of ambiguity is in the task of relation extraction, where ambiguity manifests itself in different ways of expressing the same relation.

Future directions for addressing ambiguity include the following. Historical data in the form of user or interaction logs can help to provide additional context in the case where there are ambiguities caused by several possible interpretations, e.g., different names in text. Incorporating user context will also ensure that disambiguation is performed in the correct direction, i.e., tailored towards the needs of the user. Furthermore, recent development on transformer-based models, i.e., models that consider more contextual information (i.e., long-range dependencies and word interactions) could be useful for the tasks that we have discussed (Devlin *et al.*, 2018).

We refer to *sparsity* as any situation where training data is very restricted or even unavailable, so that some feature values are rarely observed, some relationships that we observe are incomplete, or some entities are less notable. Sparsity manifests itself in many situations across tasks. In document filtering, sparsity is related to distributions of entities in the stream of documents to be filtered. Some entities appear very rarely, i.e., long-tail entities. These discrepancies in distributions would complicate the tasks as it would be easier to adjust the task to suit the more popular entities. Another example is entity typing, when some types may rarely occur and only includes a small number of entity instances in the KG. Finally, in knowledge base completion, many relations stored may actually be incomplete or missing. This further complicates the completion task, as very little training data would be available to train a reliable model. The performance of embedding-based

approaches to link prediction has been demonstrated to degrade due to sparsity (Pujara *et al.*, 2017; Zhang *et al.*, 2019). When dealing with entity-related textual data, sparsity is an issue because this limitation will affect the performance of the method. In the applications section, we presented an overview of end-to-end knowledge graph construction. The sparsity issue poses even more challenges when attempting to construct domain-specific knowledge graphs.

In future work, it would be interesting to combine different strategies to address sparsity. For example, distant supervision can be combined with techniques from other domains, e.g., signal approximation (Jin *et al.*, 2014). Training data that was generated automatically using distant supervision strategies can provide known attribute values which might inform the value of unknown attributes. Alternatively, enrichment techniques, e.g., complementing structured data in the KG with unstructured text, has also demonstrated to be effective for the task of learning entity embeddings (Kong *et al.*, 2019). Enrichment can also be performed by inferring new information. Some approaches rely on learning a set of rules or axioms to reduce sparsity, and then inferring new information based on the learned axioms and incorporating them as additional input during training (Zhang *et al.*, 2019). Furthermore, to reduce sparsity during the initial construction of KGs, a given set of entities can be prioritized instead of dividing attention over all detected entities (Hegde and Talukdar, 2015).

We refer to *temporality* as any situation where time should be considered as a separate dimension or distinct part of the task, e.g., when predictions will change over time. Many relationship types (e.g., employment, education) have a natural temporal beginning and end. Properties of an entity change over time, i.e., the popularity of a person entity in a general purpose KG might fluctuate. Models are rarely static; new facts and new entities emerge (Frank *et al.*, 2014) over time. New associations between an entity and a new topic might appear (or disappear) over time (Graus *et al.*, 2016). Furthermore, when dealing with textual data, the meaning of words and and their associated concepts could change over time (Kenter *et al.*, 2015). Therefore, for some tasks we can not simply assume that the world is static.

One way to make temporally-aware predictions is by incorporating

temporal information as features in the prediction task. In future work, temporality can also be addressed by directly taking it into account in the model. One way to achieve this is by incorporating it in a loss function that is going to be optimized. For instance, temporal information can be incorporated into an embedding-based link prediction model by modifying the scoring function (Leblay and Chekol, 2018; Dasgupta *et al.*, 2018). Alternatively, Garcia-Duran *et al.* (2018) learn time-aware representation of relation types, which is later combined with a base (i.e., non-time-aware) method.

Next, we define *explainability* as the requirement for a model to explain any prediction made in the context of a task. Fang *et al.* (2011) are the first to introduce the notion of explainability in an entity-related search setting. There are several tasks that would definitely benefit from having explainability including entity retrieval, related entity ranking, entity recommendation, etc. When using related entities to explore document collections in a particular domain, the user would want to know the rationale behind a particular ranking, to help decide when to incorporate them into their analysis (Bron *et al.*, 2010). Some use-cases in specific domains require that the ranking of related entities or recommended items is explainable. This explanation is particularly important in the case when the results of the method will be used for supporting a business decision. In some cases, a detailed explanation of how a model comes to a certain decision in every step needs to be produced. Along this line, Wang *et al.* (2018) present a recommendation method that is able to present the paths between entities as explanations.

Future directions to address explainability would be to provide more user-friendly, automatically generated explanations for users. Voskarides *et al.* (2017) show that automatically generated textual summaries that can be directly consumed by users would be quite useful. Another direction would be to learn an explanation model that would be able to explain the output of a core model that was not built with explainability in mind (Ribeiro *et al.*, 2016).

Finally, we consider *user behavior* as another important concern. Many of the gains in effectiveness for document retrieval over the past decade are due to improved ways of interpreting and learning from behavioral data. An area that is strongly under-explored concerns

the modeling of and learning from human interactions with KGs in an IR setting, in a similar vein as done with traditional document rankings (Chuklin *et al.*, 2015). Can we learn to predict interaction behavior on complex SERPs that combine organic results, direct answers, and entity knowledge cards? Either explicitly by designing suitable graphical models (Xie *et al.*, 2018) or implicitly by directly learning from interaction logs (Borisov *et al.*, 2016)? Attention patterns are likely to be different from attention patterns on traditional SERPs, leading to changes in preferred presentation order (Oosterhuis and de Rijke, 2018). Much work remains to be done.

Acknowledgements

We thank our colleagues Laura Dietz, Alexander Kotov, and Nikos Voskarides for valuable feedback and inspiration. We are very grateful to our anonymous reviewers, who provided extensive feedback, comments, and suggestions. We also thank our editors, Yiqun Liu and Mark Sanderson, for support and valuable feedback.

This research was supported by the Innovation Center for Artificial Intelligence (ICAI).

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Appendices

A

Acronyms used

ACE	Automatic Content Extraction
CBOW	continuous bag-of-words
CoNLL	Computational Natural Language Learning
CPS	center-piece subgraph
CTR	click-through rate
DCG	discounted cumulative gain
EDRM	Entity-Duet neural Ranking Model
ELR	entity linking incorporated retrieval
EM	expectation maximization
EQFE	entity query feature expansion
ERR	expected reciprocal rank
ESR	Explicit Semantic Ranking
FSDM	fielded sequential dependence model
IDF	inverse document frequency
ILP	integer linear programming
INEX	Initiative for the Evaluation of XML Retrieval
IR	information retrieval
KB	knowledge base

KBA Knowledge Base Acceleration
KBP Knowledge Base Population
KG knowledge graph
LDA latent Dirichlet allocation
LLR Log Likelihood Ratio
LSE Latent Semantic Entities
LSI Latent Semantic Indexing
MAP mean average precision
MART multiple additive regression trees
MIP mixed integer program
MLE Maximum Likelihood Estimation
MLP multi-layer perceptron
MRF Markov Random Field
MRR mean reciprocal rank
MUC Message Understanding Conference
NDCG normalized discounted cumulative gain
NER named entity recognition
NNPLB "No Noun Phrase Left Behind"
NTCIR NII Testbeds and Community for Information access Research
NVSM neural vector space model
PMI Pointwise Mutual Information
PRA path ranking algorithm
REF Related Entity Finding
RER relative error reduction
RNN recurrent neural network
SDM sequential dependence model
SE structured embedding
SELM Semantics Enabled Language Model
SERP search engine result page
SME semantic matching energy
SRDP serendipity
TAC Text Analytics Conference

TDT Topic Detection and Tracking
TF term frequency
TREC Text Retrieval Conference

B

Resources

In this appendix we list corpora, KGs, datasets, code bases, libraries, and tutorials that may help readers of this survey to further explore the area of IR and KGs.

B.1 Corpora

Corpora relevant to the work surveyed in this paper are listed in Table B.1.

B.2 Knowledge graphs

Publicly-available KGs are listed in Table B.2.

B.3 Datasets

Existing datasets relevant to the work surveyed in this paper are listed in Table B.3 and Table B.4. Table B.3 lists datasets regarding the use of KGs for IR; we use the same order of the tasks as in Chapter 4.

For *entity linking*, many datasets are available. Initially, MSNC (Cucerzan, 2007), AQUAINT (Milne and Witten, 2008), and IITB

Table B.1: Corpora commonly used in entity-related experiments.

Collection	URL
AP	https://catalog.ldc.upenn.edu/LDC93T3B
AQUAINT	https://tac.nist.gov//data/data_desc.html
ClueWeb09	https://www.lemurproject.org/clueweb09.php
ClueWeb12 (Gabrilovich *et al.*, 2013)	https://www.lemurproject.org/clueweb12.php/
GOV2	http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm
MSNBC (Cucerzan, 2007)	https://kdd.ics.uci.edu/databases/msnbc/msnbc.data.html
NYT	https://catalog.ldc.upenn.edu/ldc2008t19
TREC Robust	https://trec.nist.gov/data/t13_robust.html
WT10G	http://ir.dcs.gla.ac.uk/test_collections/wt10g.html

Table B.2: General-purpose and domain-specific knowledge graphs.

Knowledge Graph	URL
DBpedia (Lehmann *et al.*, 2015)	https://wiki.dbpedia.org
Freebase	http://freebase.org
Wikipedia	http://wikipedia.org
YAGO (Suchanek *et al.*, 2007)	https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/
LittleSis	http://littlesis.org

(Kulkarni *et al.*, 2009) were introduced. Hoffart *et al.* (2011) created a dataset based on CoNLL 2003, annotating proper nouns with corresponding entities in YAGO. Meij *et al.* (2012) introduced a dataset for experimenting with entity linking in tweets. GERBIL (Usbeck *et al.*, 2015) provides a common service to evaluate entity linking systems. Carmel *et al.* (2014) organize the 2014 Entity Recognition and Disambiguation Challenge. Unlike other entity linking dataset, the mention segmentation were not given in this challenge. Hasibi *et al.* (2015) specify entity linking tasks in the context of query understanding.

Evaluation of document retrieval can be performed without any

entity-specific evaluation dataset. Most of the research work utilizing entity information such as (Dalton *et al.*, 2014; Raviv *et al.*, 2016) use the TREC Robust collection (Voorhees, 2005). The dataset in (Balog *et al.*, 2009b) can be used to evaluate *entity recommendations*. No recent entity recommendation test collections have been released.

One of the earliest entity retrieval test collection is INEX-ER (de Vries *et al.*, 2008). Later on, Balog and Ramampiaro (2013) develop an *entity retrieval* test collection based on the DBpedia knowledge base. They combine queries from the following evaluation campaigns: INEX-ER (entity ranking), TREC Entity (related entity finding task), SemSearch ES (ad-hoc entity search), SemSearch LS (searching lists of entity), QALD-2 (question answering over linked data), and finally INEX-LD (ad-hoc search). In the context of query understanding Schuhmacher *et al.* (2015) collected relevance judgments of entities with respect to queries.

On the task of *relationship explanation*, (Voskarides *et al.*, 2015) introduce the first relation explanation dataset based on sentences extracted from Wikipedia articles.

Table B.3: Datasets used in tasks related to leveraging KGs for IR.

Task	Datasets
Entity linking	MSNBC (Cucerzan, 2007)
	AQUAINT (Milne and Witten, 2008)
	IITB (Kulkarni *et al.*, 2009)
	AIDA CoNLL-YAGO (Hoffart *et al.*, 2011)
	Twitter-to-concept (Meij *et al.*, 2012)
	FACC (Gabrilovich *et al.*, 2013)
	GERBIL (Usbeck *et al.*, 2015)
	Wikinews/Meantime (Minard *et al.*, 2016)
	GERDAQ4 (Cornolti *et al.*, 2016)
	ERD (Carmel *et al.*, 2014)
	Wikilinks (Singh *et al.*, 2012)
Document retrieval	TREC Robust (Voorhees, 2005)
Entity recommendation	REF (Balog *et al.*, 2009b)
Entity retrieval	INEX-ER (de Vries *et al.*, 2008)
	DBpedia-Entity (Balog and Neumayer, 2013)
	REWQ (Schuhmacher *et al.*, 2015)
Relationship explanation	(Voskarides *et al.*, 2015)

Table B.4 details datasets related to tasks dealing with *information retrieval for knowledge graphs*. Early research on *named entity recognition* is mainly driven by the CoNLL evaluation campaign. This dataset employs the BIO labeling scheme (i.e., indicating beginning, inside, and outside of entity segment) to indicate entity segment boundaries in the text (Tjong Kim Sang and De Meulder, 2003). Later, Hachey *et al.* (2014) proposed a shared evaluation paradigm for the task of entity recognition and disambiguation. The evaluation software and standardized system outputs are provided online.[1] This dataset can be used to evaluate the mention detection (i.e., entity recognition) and disambiguation in an end-to-end fashion.

Entity discovery was officially introduced as a new variant in TAC KBP 2014 (Ellis *et al.*, 2014). Annotators exhaustively annotated all named named mentions of persons, organizations, and locations occurred in documents.

As for *entity typing*, no publicly available dataset especially designed for the tasks exist, however researchers can improvise by using known types in existing KGs in their experiments (see Table B.2).

The task of *entity-centric document filtering* was introduced as part of the TREC KBA evaluation campaign (Frank *et al.*, 2014), which ran for three years. The last year of the evaluation campaign (2014) presented queries and annotations, including a subset containing documents where query entities are mentioned, making it easier for researchers in this area.

One of earliest datasets for *relation extraction* was introduced in the Automatic Content Extractions campaign (Doddington *et al.*, 2004). As for knowledge base completion, Bordes *et al.* (2013) introduced FB15K, a subset extracted from Freebase; Toutanova and Chen (2015) later on extend this dataset. In the context of KG *quality control*, Heindorf *et al.*, 2015 construct an interesting dataset for vandalism detection in knowledge bases, built based on the revision history of Wikidata.

[1] https://github.com/wikilinks/neleval

Table B.4: Datasets used in tasks related to constructing KGs.

Task	Datasets
Entity recognition	CoNLL (Tjong Kim Sang and De Meulder, 2003)
Entity-centric document filtering	TREC-KBA StreamCorpus (Frank *et al.*, 2014)
Entity discovery	TAC-KBP EDL (Ellis *et al.*, 2014)
Relation Extraction	SemEval (Girju *et al.*, 2007)
Link prediction	ACE (Doddington *et al.*, 2004) WN18 (Bordes *et al.*, 2013) FB15K (Bordes *et al.*, 2013) FB15K-237 (Toutanova and Chen, 2015)
KG quality estimation	WDVC (Heindorf *et al.*, 2015)

B.4 Code

In Table B.5 we list publicly available implementations of entity-related systems surveyed in this paper. Most of these are original implementations of the work.

B.5 Libraries

We list publicly available implementations of related libraries that might be required for entity-related experiments in Table B.6 and Table B.7.

B.6 Tutorials

Finally, recent tutorials in the area of IR and KGs are listed in Table B.8, with the most recent version cited. First, we list "An introduction to entity recommendation and understanding" (Ma and Ke, 2015). This tutorial presents an introduction and overview of emerging topics in entity recommendation and understanding. Starting with a basic introduction on KGs, and finally diving deeper into various recommendation algorithms. In "Constructing and mining web-scale knowledge graphs," Gabrilovich and Usunier (2016) present the state of the art in constructing, mining, and growing KGs. The authors give the basic concepts, tools and methodologies, datasets, and open research challenges. "Entity

Table B.5: Entity-related systems.

Implementation	URL
Entity linking	
AIDA (Hoffart *et al.*, 2011)	https://github.com/codepie/aida
DBpedia spotlight (Daiber *et al.*, 2013)	https://github.com/dbpedia-spotlight/dbpedia-spotlight
Illinois wikifier (Ratinov *et al.*, 2011)	https://cogcomp.org/page/software_view/Wikifier
Semanticizer (Odijk *et al.*, 2013)	http://semanticize.uva.nl
TagME (Ferragina and Scaiella, 2010)	https://github.com/marcocor/tagme-python
Wikipedia miner (Milne and Witten, 2008)	https://github.com/dnmilne/wikipediaminer/wiki/About-wikipedia-miner
PBOL (Ganea *et al.*, 2015)	https://github.com/dalab/pboh-entity-linking
SMAPH (Cornolti *et al.*, 2016)	https://github.com/marcocor/smaph
Entity retrieval	
SERT (Van Gysel *et al.*, 2017a)	https://github.com/cvangysel/SERT
FieldedSDM (Zhiltsov *et al.*, 2015)	https://github.com/teanalab/FieldedSDM
PSDM (Nikolaev *et al.*, 2016)	https://github.com/teanalab/pfsdm
Entity recognition	
Stanford NER (Finkel *et al.*, 2005)	https://nlp.stanford.edu/software/CRF-NER.html
ClusType (Ren *et al.*, 2015)	https://github.com/shanzhenren/ClusType
FIGER (Ling and Weld, 2012)	https://github.com/xiaoling/figer
AFET (Ren *et al.*, 2016a)	https://github.com/shanzhenren/AFET
KG embedding	
OpenKE	http://openke.thunlp.org/home

linking and retrieval for semantic search" (Meij *et al.*, 2014) provides a comprehensive overview of entity linking and retrieval in the context of semantic search, including query understanding and entity retrieval and

Table B.6: Document retrieval systems.

Library	URL
Indri/Lemur	https://www.lemurproject.org/indri.php
Lucene	http://lucene.apache.org/
Terrier	http://terrier.org

Table B.7: Learning to rank systems.

Library	URL
JForests (Ganjisaffar *et al.*, 2011)	https://github.com/yasserg/jforests
RankLib	https://sourceforge.net/p/lemur/wiki/RankLib/

ranking techniques on structured data. Finally, in "Utilizing Knowledge Graphs in Text-centric Information Retrieval," Dietz *et al.* (2017) give a brief overview of different types of knowledge bases, ad-hoc object retrieval, and also entity linking techniques.

Table B.8: Tutorials in the area of IR and KGs.

Tutorial	URL
An introduction to entity recommendation and understanding	https://www.microsoft.com/en-us/research/publication/an-introduction-to-entity-recommendation-and-understanding
Constructing and mining web-scale knowledge graphs	http://www.cs.technion.ac.il/~gabr/publications/papers/KDD14-T2-Bordes-Gabrilovich.pdf
Entity linking and retrieval for semantic search	https://github.com/ejmeij/entity-linking-and-retrieval-tutorial
Utilizing Knowledge Graphs in Text-centric Information Retrieval	https://github.com/laura-dietz/tutorial-kb4ir

References

Abhishek, A., A. Anand, and A. Awekar (2017). "Fine-Grained Entity Type Classification by Jointly Learning Representations and Label Embeddings". In: *EACL '17*. ACL.

Agichtein, E. and L. Gravano (2000). "Snowball: Extracting Relations from Large Plain-text Collections". In: *DL '00*. ACM.

Alfonseca, E., K. Filippova, J.-Y. Delort, and G. Garrido (2012). "Pattern Learning for Relation Extraction with a Hierarchical Topic Model". In: *ACL '12*. *ACL*.

Alhelbawy, A. and R. Gaizauskas (2014). "Graph Ranking for Collective Named Entity Disambiguation". In: *ACL '14*. ACL.

Allan, J. (2002). "Introduction to Topic Detection and Tracking". In: *Topic Detection and Tracking*. Kluwer Academic Publishers. 1–16.

Asahara, M. and Y. Matsumoto (2003). "Japanese Named Entity Extraction with Redundant Morphological Analysis". In: *HLT-NAACL '03*. ACL.

Bach, N. and S. Badaskar (2007). "A Survey on Relation Extraction". Language Technologies Institute, Carnegie Mellon University.

Balog, K. (2018). *Entity-Oriented Search*. Springer.

Balog, K., L. Azzopardi, and M. de Rijke (2006). "Formal Models for Expert Finding in Enterprise Corpora". In: *SIGIR '06*. ACM.

Balog, K., L. Azzopardi, and M. de Rijke (2009a). "A Language Modeling Framework for Expert Finding". *Information Processing & Management*. 45(1): 1–19.

Balog, K., M. Bron, and M. de Rijke (2011). "Query Modeling for Entity Search Based on Terms, Categories, and Examples". *ACM Transactions on Information Systems*. 29(4): 22:1–22:31.

Balog, K., Y. Fang, M. de Rijke, P. Serdyukov, and L. Si (2012). "Expertise Retrieval". *Foundations and Trends in Information Retrieval*. 6(2-3): 127–256.

Balog, K. and R. Neumayer (2013). "A Test Collection for Entity Search in DBpedia". In: *SIGIR '13*. ACM.

Balog, K. and H. Ramampiaro (2013). "Cumulative Citation Recommendation: Citation vs. Ranking". In: *SIGIR '13*. ACM.

Balog, K., H. Ramampiaro, N. Takhirov, and K. Nørvåg (2013). "Multi-step Classification Approaches to Cumulative Citation Recommendation". In: *OAIR '13*. Le Centre De Hautes Etudes Internationales D'Informatique Documentaire. 121–128.

Balog, K., P. Serdyukov, A. P. de Vries, P. Thomas, and T. Westerveld (2009b). "Overview of the TREC 2009 Entity Track". In: *TREC*.

Banko, M., M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni (2007). "Open Information Extraction from the Web". In: *IJCAI '07*. ACM.

Bao, J., N. Duan, M. Zhou, and T. Zhao (2014). "Knowledge-Based Question Answering as Machine Translation". In: *ACL '14*. ACL.

Bast, H., B. Buchhold, and E. Haussmann (2015). "Relevance Scores for Triples from Type-Like Relations". In: *SIGIR '15*. ACM.

Bast, H., B. Buchhold, and E. Haussmann (2016). "Semantic Search on Text and Knowledge Bases". *Foundations and Trends in Information Retrieval*. 10(2-3): 119–271.

Berant, J., A. Chou, R. Frostig, and P. Liang (2013). "Semantic Parsing on Freebase from Question-Answer Pairs". In: *EMNLP '13*.

Bi, B., H. Ma, B.-J. Hsu, W. Chu, K. Wang, and J. Cho (2015). "Learning to Recommend Related Entities to Search Users". In: *WSDM '15*. ACM.

Bikel, D. M., R. Schwartz, and R. M. Weischedel (1999). "An Algorithm That Learns What's in a Name". *Machine Learning*. 34(1–3): 211–231.

Bing, L., W. Lam, and T.-L. Wong (2013). "Wikipedia Entity Expansion and Attribute Extraction from the Web Using Semi-supervised Learning". In: *WSDM '13*. ACM.

Blanco, R., B. B. Cambazoglu, P. Mika, and N. Torzec (2013). "Entity Recommendations in Web Search". In: *ISWC '13*. Springer-Verlag.

Blanco, R., G. Ottaviano, and E. Meij (2015). "Fast and Space-Efficient Entity Linking for Queries". In: *WSDM '15*. ACM.

Bonnefoy, L., V. Bouvier, and P. Bellot (2013). "A Weakly-Supervised Detection of Entity Central Documents in a Stream". In: *SIGIR '13*. ACM.

Bordes, A., X. Glorot, J. Weston, and Y. Bengio (2014). "A Semantic Matching Energy Function for Learning with Multi-relational Data". *Machine Learning*. 94(2): 233–259.

Bordes, A., N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko (2013). "Translating Embeddings for Modeling Multi-relational Data". In: *NIPS '13*. JLMR.

Bordes, A., J. Weston, R. Collobert, and Y. Bengio (2011). "Learning Structured Embeddings of Knowledge Bases". In: *AAAI '11*. AAAI Press.

Bordino, I., G. De Francisci Morales, I. Weber, and F. Bonchi (2013a). "From Machu_Picchu to "rafting the urubamba river": Anticipating Information Needs via the Entity-Query Graph". In: *WSDM '13*. ACM.

Bordino, I., Y. Mejova, and M. Lalmas (2013b). "Penguins in Sweaters, or Serendipitous Entity Search on User-Generated Content". In: *CIKM '13*. ACM.

Borisov, A., I. Markov, M. de Rijke, and P. Serdyukov (2016). "A Neural Click Model for Web Search". In: *WWW '16*. ACM.

Borthwick, A. E. (1999). "A Maximum Entropy Approach to Named Entity Recognition". *PhD thesis*. New York, NY, USA: New York University.

Bota, H., K. Zhou, and J. M. Jose (2016). "Playing Your Cards Right: The Effect of Entity Cards on Search Behaviour and Workload". In: *CHIIR '16*. ACM.

Brambilla, M., S. Ceri, F. Daniel, M. Di Giovanni, A. Mauri, and G. Ramponi (2018). "Iterative Knowledge Extraction from Social Networks". In: *WWW '18 Companion*. ACM.

Brambilla, M., S. Ceri, E. Della Valle, R. Volonterio, and F. X. Acero Salazar (2017). "Extracting Emerging Knowledge from Social Media". In: *WWW '17. ACM*.

Brin, S. (1998). "Extracting Patterns and Relations from the World Wide Web". In: *WebDB '98*. Springer-Verlag.

Bron, M., K. Balog, and M. de Rijke (2010). "Ranking Related Entities: Components and Analyses". In: *CIKM '10*. ACM.

Bronnenberg, W., H. Bunt, J. Landsbergen, R. Scha, W. Schoenmakers, and E. van Utteren (1980). "The Question Answering System Phliqa1". In: *Natural Language Question Answering Systems*. Ed. by L. Bolc. MacMillan. 217–305.

Bunescu, R. C. and R. J. Mooney (2005). "A Shortest Path Dependency Kernel for Relation Extraction". In: *HTL-EMNLP '05*. ACL.

Bunescu, R. and M. Pasca (2006). "Using Encyclopedic Knowledge for Named Entity Disambiguation". In: *EACL*. ACL.

Cai, R., H. Wang, and J. Zhang (2015). "Learning Entity Representation for Named Entity Disambiguation". In: *ACL'13*. ACL.

Cano, I., S. Singh, and C. Guestrin (2014). "Distributed Non-Parametric Representations for Vital Filtering: UW at TREC KBA 2014". In: *TREC 2014*. NIST.

Carmel, D., M.-W. Chang, E. Gabrilovich, B.-J. Hsu, and K. Wang (2014). "ERD'14: Entity Recognition and Disambiguation Challenge". In: *SIGIR '14*. ACM.

Ceccarelli, D., C. Lucchese, S. Orlando, R. Perego, and S. Trani (2013). "Learning Relatedness Measures for Entity Linking". In: *CIKM '13*. ACM.

Charton, E., M.-J. Meurs, L. Jean-Louis, and M. Gagnon (2014). "Mutual Disambiguation for Entity Linking". In: *ACL '14*. *ACL*.

Chen, M., Y. Tian, M. Yang, and C. Zaniolo (2016). "Multi-lingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment". *CoRR*. abs/1611.03954.

Chuklin, A., I. Markov, and M. de Rijke (2015). *Click Models for Web Search*. Morgan & Claypool Publishers.

Collins, M. and Y. Singer (1999). "Unsupervised Models for Named Entity Classification". In: *EMNLP '99*. *ACL*.

Cornolti, M., P. Ferragina, and M. Ciaramita (2013). "A Framework for Benchmarking Entity-annotation Systems". In: *WWW '13*. ACM.

Cornolti, M., P. Ferragina, M. Ciaramita, S. Rüd, and H. Schütze (2016). "A Piggyback System for Joint Entity Mention Detection and Linking in Web Queries". In: *WWW '16*. ACM.

Corro, L. D., A. Abujabal, R. Gemulla, and G. Weikum (2015). "FINET: Context-Aware Fine-Grained Named Entity Typing". In: *EMNLP '15*. ACL.

Craswell, N., A. P. de Vries, and I. Soboroff (2005). "Overview of the TREC-2005 enterprise track". In: *TREC 2005 Conference Notebook*. NIST.

Croft, B., D. Metzler, and T. Strohman (2009). *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company.

Cucchiarelli, A. and P. Velardi (2001). "Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence". *Computational Linguistics*. 27(1): 123–131.

Cucerzan, S. (2007). "Large-Scale Named Entity Disambiguation Based on Wikipedia Data". In: *EMNLP '07*. ACL.

Culotta, A. and J. Sorensen (2004). "Dependency Tree Kernels for Relation Extraction". In: *ACL '04*. ACL.

Daiber, J., M. Jakob, C. Hokamp, and P. N. Mendes (2013). "Improving Efficiency and Accuracy in Multilingual Entity Extraction". In: *Semantic Systems '13*.

Dalton, J., J. Allan, and D. A. Smith (2011). "Passage Retrieval for Incorporating Global Evidence in Sequence Labeling". In: *CIKM '11*. ACM.

Dalton, J. and L. Dietz (2013). "Constructing Query-specific Knowledge Bases". In: *AKBC '13*. ACM.

Dalton, J., L. Dietz, and J. Allan (2014). "Entity Query Feature Expansion Using Knowledge Base Links". In: *SIGIR '14*. ACM.

Dalvi, B., A. Mishra, and W. W. Cohen (2016). "Hierarchical Semi-supervised Classification with Incomplete Class Hierarchies". In: *WSDM '16*. *ACM*.

Dasgupta, S. S., S. N. Ray, and P. P. Talukdar (2018). "HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding". In: *EMNLP '18*. ACL.

de Vries, A. P., A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas (2008). "Overview of the INEX 2007 Entity Ranking Track". In: *Focused Access to XML Documents*. Ed. by N. Fuhr, J. Kamps, M. Lalmas, and A. Trotman. Springer-Verlag.

Devlin, J., M. Chang, K. Lee, and K. Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding".

Dietz, L. and J. Dalton (2013). "UMass at TREC 2013 Knowledge Base Acceleration track". In: *TREC 2013*. NIST.

Dietz, L., A. Kotov, and E. Meij (2017). "Utilizing Knowledge Graphs in Text-centric Information Retrieval". In: *WSDM '17*. ACM.

Dietz, L., A. Kotov, and E. Meij (2018). "Utilizing Knowledge Graphs for Text-Centric Information Retrieval". In: *SIGIR '18*. ACM.

Doddington, G., A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel (2004). "The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation". In: *LREC '04*. ACL.

Dong, L., F. Wei, H. Sun, M. Zhou, and K. Xu (2015a). "A Hybrid Neural Model for Type Classification of Entity Mentions". In: *IJCAI '15*. AAAI Press.

Dong, X. L., E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang (2014a). "From Data Fusion to Knowledge Fusion". In: *VLDB '14*. VLDB Endowment.

Dong, X. L., E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang (2015b). "Knowledge-based Trust: Estimating the Trustworthiness of Web Sources". In: *VLDP '15*. VLDB Endowment.

Dong, X., E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang (2014b). "Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion". In: *KDD '14*. ACM.

Dutta, S. and G. Weikum (2015). "C3EL: A Joint Model for Cross-Document Co-Reference Resolution and Entity Linking". In: *EMNLP '15*. ACL.

Efron, M., C. Willis, and G. Sherman (2014). "Learning Sufficient Queries for Entity Filtering". In: *SIGIR '14*. ACM.

Ellis, J., J. Getman, and S. Strassel (2014). "Overview of Linguistic Resources for the TAC KBP 2014 Evaluations: Planning, Execution, and Results". In: *TAC '14*. LDC.

Ensan, F. and E. Bagheri (2017). "Document Retrieval Model Through Semantic Linking". In: *WSDM '17*.

Etzioni, O., M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates (2004). "Web-scale Information Extraction in Knowitall: (Preliminary Results)". In: *WWW '04*. ACM.

Etzioni, O., A. Fader, J. Christensen, S. Soderland, and Mausam (2011). "Open Information Extraction: The Second Generation". In: *IJCAI '11*. AAAI Press.

Fader, A., S. Soderland, and O. Etzioni (2011). "Identifying Relations for Open Information Extraction". In: *EMNLP '11*. ACL.

Fang, L., A. D. Sarma, C. Yu, and P. Bohannon (2011). "REX: Explaining Relationships Between Entity Pairs". In: *VLDB '11*. VLDB Endowment.

Fathalla, S., S. Vahdati, S. Auer, and C. Lange (2017). "Towards a Knowledge Graph Representing Research Findings by Semantifying Survey Articles". In: *TPDL '17*. Springer-Verlag.

Ferragina, P. and U. Scaiella (2010). "TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities)". In: *CIKM '10*. ACM.

Fetahu, B., K. Markert, and A. Anand (2015). "Automated News Suggestions for Populating Wikipedia Entity Pages". In: *CIKM '15*. ACM.

Finkel, J. R., T. Grenager, and C. Manning (2005). "Incorporating Non-local Information Into Information Extraction Systems by Gibbs sampling". In: *ACL '05*. ACL.

Fissaha Adafre, S. and M. de Rijke (2007). "Ask the crowd to find out what's important". In: *ICDM'07 Workshop on Data Mining in Web 2.0 Environments*.

Flekova, L., O. Ferschke, and I. Gurevych (2014). "What Makes a Good Biography?" In: *WWW '14*. ACM.

Foley, J., B. O'Connor, and J. Allan (2016). "Improving Entity Ranking for Keyword Queries". In: *CIKM '16*. ACM.

Franco-Salvador, M., P. Rosso, and R. Navigli (2014). "A Knowledge-based Representation for Cross-Language Document Retrieval and Categorization". In: *EACL '14*. ACL.

Frank, J. R., M. Kleiman-Weiner, D. A. Roberts, F. Niu, Z. Ce, R. Christopher, and I. Soboroff (2012). "Building an Entity-Centric Stream Filtering Test Collection for TREC 2012". In: *TREC 2012*. NIST.

Frank, J. R., M. Kleiman-Weiner, D. A. Roberts, E. Voorhees, and I. Soboroff (2014). "TREC KBA Overview". In: *TREC 2014*. NIST.

Fuxman, A. (2015). "In Situ Insights". In: *SIGIR '15*. 655–664.

Gabrilovich, E. and S. Markovitch (2007). "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis". In: *IJCAI '07*. AAAI Press.

Gabrilovich, E. and N. Usunier (2016). "Constructing and Mining Web-scale Knowledge Graphs". In: *SIGIR '16*. ACM.

Gabrilovich, E., M. Ringaard, and A. Subramanya (2013). "FACC1: Freebase annotation of ClueWeb corpora, Version 1". *Tech. rep.* Google Research.

Galárraga, L. A., N. Preda, and F. M. Suchanek (2013). "Mining Rules to Align Knowledge Bases". In: *AKBC '13*. ACM.

Ganea, O.-E., M. Horlescu, A. Lucchi, C. Eickhoff, and T. Hofmann (2015). "Probabilistic Bag-Of-Hyperlinks Model for Entity Linking". In: *WWW '16*. ACM.

Ganjisaffar, Y., R. Caruana, and C. Lopes (2011). "Bagging Gradient-Boosted Trees for High Precision, Low Variance Ranking Models". In: *SIGIR '11*. ACM.

Gao, J., P. Pantel, M. Gamon, X. He, and L. Deng (2014). "Modeling Interestingness with Deep Neural Networks". In: *EMNLP '14*. ACL.

Gao, J., X. Li, Y. E. Xu, B. Sisman, X. L. Dong, and J. Yang (2019). "Efficient Knowledge Graph Accuracy Evaluation". In: *VLDB '19*. VLDB Endowment.

Garcia-Duran, A., S. Dumancic, and M. Niepert (2018). "Learning Sequence Encoders for Temporal Knowledge Graph Completion". In: *EMNLP '18*.

Gardner, M. and T. Mitchell (2015). "Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction". In: *EMNLP '15*. ACL. 1488–1498.

Gardner, M., P. P. Talukdar, B. Kisiel, and T. Mitchell (2013). "Improving Learning and Inference in a Large Knowledge-base using Latent Syntactic Cues". In: *EMNLP '13*. ACL.

Gerritse, E. J., F. Hasibi, and A. P. de Vries (2020). "Graph-Embedding Empowered Entity Retrieval". In: *Advances in Information Retrieval*. Ed. by J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, and F. Martins. Cham: Springer International Publishing. 97–110.

Gillick, D., N. Lazic, K. Ganchev, J. Kirchner, and D. Huynh (2014). "Context-Dependent Fine-Grained Entity Type Tagging". *CoRR*.

Girju, R., P. Nakov, V. Nastase, S. Szpakowicz, P. Turney, and D. Yuret (2007). "SemEval-2007 Task 04: Classification of Semantic Relations Between Nominals". In: *SemEval '07*. ACL.

Globerson, A., N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringgaard, and F. Pereira (2016). "Collective Entity Resolution with Multi-Focal Attention". In: *ACL '16*. ACM.

Graus, D., D. Odijk, and M. de Rijke (2018). "The Birth of Collective Memories: Analyzing Emerging Entities in Text Streams". *Journal of the Association for Information Science and Technology*. 69(6): 773–786.

Graus, D., M. Tsagkias, L. Buitinck, and M. de Rijke (2014). "Generating pseudo-ground truth for predicting new concepts in social streams". In: *ECIR '14*. Springer-Verlag.

Graus, D., M. Tsagkias, W. Weerkamp, E. Meij, and M. de Rijke (2016). "Dynamic Collective Entity Representations for Entity Ranking". In: *WSDM '16*. ACM.

Green, B. F., A. K. Wolf, C. Chomsky, and K. Laughery (1963). "Baseball: An Automatic Question Answerer". In: *Computers and Thought*. Ed. by E. Figenbaum and J. Fledman. McGraw-Hill. 207–216.

Grishman, R. and B. Sundheim (1996). "Message Understanding Conference-6: A Brief History". In: *COLING '96*. ACL.

Guo, Y., B. Qin, T. Liu, and S. Li (2013). "Microblog Entity Linking by Leveraging Extra Posts". In: *EMNLP '13*. ACL.

GuoDong, Z., S. Jian, Z. Jie, and Z. Min (2005). "Exploring Various Knowledge in Relation Extraction". In: *ACL '05*. ACL.

Gupta, N., S. Singh, and D. Roth (2017). "Entity Linking via Joint Encoding of Types, Descriptions, and Context". In: *EMNLP '17*. ACL.

Hachey, B., J. Nothman, and W. Radford (2014). "Cheap and easy entity evaluation". In: *ACL '14*. ACL.

Hakkani-Tür, D., A. Celikyilmaz, L. Heck, G. Tur, and G. Zweig (2014). "Probabilistic Enrichment of Knowledge Graph Entities for Relation Detection in Conversational Understanding". In: *Interspeech 2014*. ACM.

Hasibi, F., K. Balog, and S. E. Bratsberg (2015). "Entity Linking in Queries: Tasks and Evaluation". In: *ICTIR '15*. ACM.

Hasibi, F., K. Balog, and S. E. Bratsberg (2016). "Exploiting Entity Linking in Queries for Entity Retrieval". In: *ICTIR '16*. ACM.

Hegde, M. (2015). "An Entity-centric Approach for Overcoming Knowledge Graph Sparsity". In: *EMNLP '15*. ACL.

Hegde, M. and P. P. Talukdar (2015). "An Entity-centric Approach for Overcoming Knowledge Graph Sparsity". In: *EMNLP '15*. ACL.

Heindorf, S., M. Potthast, B. Stein, and G. Engels (2015). "Towards Vandalism Detection in Knowledge Bases: Corpus Construction and Analysis". In: *SIGIR '15*. *ACM*.

Heindorf, S., M. Potthast, B. Stein, and G. Engels (2016). "Vandalism Detection in Wikidata". In: *CIKM '16*. ACM.

Hoffart, J., Y. Altun, and G. Weikum (2014). "Discovering Emerging Entities with Ambiguous Names". In: *WWW '14*. ACM.

Hoffart, J., D. Milchevski, G. Weikum, A. Anand, and J. Singh (2016). "The Knowledge Awakens: Keeping Knowledge Bases Fresh with Emerging Entities". In: *WWW '16 Companion*. ACM.

Hoffart, J., M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum (2011). "Robust Disambiguation of Named Entities in Text". In: *EMNLP '11*. ACL.

Hoffmann, R., C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld (2011). "Knowledge-based Weak Supervision for Information Extraction of Overlapping Relations". In: *ACL-HLT '11*. ACL.

Hovy, D. (2014). "How Well Can We Learn Interpretable Entity Types from Text?" In: *ACL '14*. ACL.

Jatowt, A. and S. Yamamoto (2017). "Overview of NTCIR-13 Actionable Knowledge Graph (AKG) Task". In: *NCTIR*. NTCIR.

Jenatton, R., N. L. Roux, A. Bordes, and G. Obozinski (2012). "A Latent Factor Model for Highly Multi-relational Data". In: *NIPS '12*. JLMR.

Ji, G., S. He, L. Xu, K. Liu, and J. Zhao (2015). "Knowledge Graph Embedding via Dynamic Mapping Matrix". In: *ACL '15*. ACL.

Ji, H., R. Grishman, and H. T. Dang (2011). "Overview of the TAC 2011 Knowledge Base Population Task". In: *Proceedings of the TAC-KBP 2011 Workshop*. NIST.

Jiang, J. and C.-y. Lin (2014). "MSR KMG at TREC 2014 KBA Track Vital Filtering Task". In: *TREC 2014*. NIST.

Jin, Y., E. Kıcıman, K. Wang, and R. Loynd (2014). "Entity Linking at the Tail: Sparse Signals, Unknown Entities, and Phrase Models". In: *WSDM '14*. ACM.

Joshi, M., U. Sawant, and S. Chakrabarti (2014). "Knowledge Graph and Corpus Driven Segmentation and Answer Inference for Telegraphic Entity-seeking Queries". In: *EMNLP*. ACL. 1104–1114.

Kambhatla, N. (2004). "Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations". In: *ACL '04*. ACL.

Kang, C., S. Vadrevu, R. Zhang, R. van Zwol, L. G. Pueyo, N. Torzec, J. He, and Y. Chang (2011). "Ranking Related Entities for Web Search Queries". In: *WWW '11*. ACM.

Kenter, T., M. Wevers, P. Huijnen, and M. de Rijke (2015). "Ad Hoc Monitoring of Vocabulary Shifts over Time". In: *CIKM '15*. ACM.

Kim, J., X. Xue, and W. B. Croft (2009). "A Probabilistic Retrieval Model for Semistructured Data". In: *ECIR '09*. Springer-Verlag.

Kipf, T. N. and M. Welling (2016). "Semi-supervised classification with graph convolutional networks". *arXiv preprint arXiv:1609.02907*.

Kolitsas, N., O.-E. Ganea, and T. Hofmann (2018). "End-to-End Neural Entity Linking". In: *CoNLL '18*.

Kong, F., R. Zhang, H. Guo, S. Mensah, Z. Hu, and Y. Mao (2019). "A Neural Bag-of-Words Modelling Framework for Link Prediction in Knowledge Bases with Sparse Connectivity". In: *WWW '19*. ACM.

Kotnis, B., P. Bansal, and P. Talukdar (2015). "Knowledge Base Inference using Bridging Entities". In: *EMNLP '15*. ACL.

Kripke, S. (1980). *Naming and Necessity*. Wiley-Blackwell.

Krishna, R., Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei (2016). "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations". *arXiv preprint arxiv:1602.07332*. Feb.

Kulkarni, S., A. Singh, G. Ramakrishnan, and S. Chakrabarti (2009). "Collective Annotation of Wikipedia Entities in Web Text". In: *KDD '09*. ACM.

Kuzey, E., J. Vreeken, and G. Weikum (2014). "A Fresh Look on Knowledge Bases: Distilling Named Events from News". In: *CIKM '14*. ACM.

Lagun, D., C.-H. Hsieh, D. Webster, and V. Navalpakkam (2014). "Towards Better Measurement of Attention and Satisfaction in Mobile Search". In: *SIGIR '14*. ACM.

Lample, G., M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer (2016). "Neural Architectures for Named Entity Recognition". In: *HLT-NAACL '16*. ACL. 260–270.

Lao, N. and W. W. Cohen (2010). "Relational Retrieval Using a Combination of Path-constrained Random Walks". *Machine Learning*. 81(1): 53–67.

Lazic, N., A. Subramanya, M. Ringgaard, and F. Pereira (2015). "Plato: A Selective Context Model for Entity Resolution". *Transactions of the Association for Computational Linguistics*. 3: 503–515.

Leblay, J. and M. W. Chekol (2018). "Deriving Validity Time in Knowledge Graph". In: *WWW '18*. ACM.

Lee, J., A. Fuxman, B. Zhao, and Y. Lv (2015). "Leveraging Knowledge Bases for Contextual Entity Exploration". In: *KDD '15*. ACM.

Lehmann, J., R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer (2015). "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia". *Semantic Web Journal*. 6(2): 167–195.

Li, F., X. Dong, A. Langen, and Y. D. Li (2017). "Knowledge Verification for LongTail Verticals".

Li, X., G. Tur, D. Hakkani-Tür, and Q. Li (2014). "Personal Knowledge Graph Population from User Utterances in Conversational Understanding". In: *SLT '14*. IEEE.

Li, X., J. Tang, T. Wang, Z. Luo, and M. de Rijke (2015). "Automatically Assessing Wikipedia Article Quality by Exploiting Article-Editor Networks". In: *ECIR '15*. Springer-Verlag.

Liao, Z., X. Song, Y. Shen, S. Lee, J. Gao, and C. Liao (2017). "Deep Context Modeling for Web Query Entity Disambiguation". In: *CIKM '17*. ACM.

Lin, T., Mausam, and O. Etzioni (2012). "No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities". In: *EMNLP-CoNLL '12*. ACL.

Lin, Y., Z. Liu, M. Sun, Y. Liu, and X. Zhu (2015). "Learning Entity and Relation Embeddings for Knowledge Graph Completion". In: *AAAI '15*. AAAI Press.

Ling, X. and D. S. Weld (2012). "Fine-grained Entity Recognition". In: *AAAI '12*. AAAI Press.

Liu, Q., L. Jiang, M. Han, Y. Liu, and Z. Qin (2016). "Hierarchical Random Walk Inference in Knowledge Graphs". In: *SIGIR '16*. ACM.

Liu, X., J. Darko, and H. Fang (2013). "A Related Entity based Approach for Knowledge Base Acceleration". In: *TREC 2013*. NIST.

Liu, X. and H. Fang (2015). "Latent Entity Space: A Novel Retrieval Approach for Entity-Bearing Queries". *Information Retrieval Journal*. 18(6): 473–503.

Liu, Z., C. Xiong, M. Sun, and Z. Liu (2018). "Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval". In: *ACL '18*. ACL.

Luo, G., X. Huang, C.-Y. Lin, and Z. Nie (2015a). "Joint Entity Recognition and Disambiguation". In: *EMNLP '15*. ACL.

Luo, Y., Q. Wang, B. Wang, and L. Guo (2015b). "Context-Dependent Knowledge Graph Embedding". In: *EMNLP '15*. ACL.

Ma, H. and Y. Ke (2015). "An Introduction to Entity Recommendation and Understanding". In: *WWW' 15 Companion*. ACM.

Ma, W., M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, and X. Ren (2019). "Jointly Learning Explainable Rules for Recommendation with Knowledge Graph". In: *WWW '19*. ACM.

Manning, C. D., P. Raghavan, and H. Schütze (2008). *Introduction to Information Retrieval*. Cambridge University Press.

McCallum, A. and W. Li (2003). "Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons". In: *CONLL '03*. ACL.

Medelyan, O., I. H. Witten, and D. Mile (2008). "Topic Indexing with Wikipedia". In: *WIKAI '08*.

Meij, E., K. Balog, and D. Odijk (2013). "Entity Linking and Retrieval". In: *SIGIR '13*. ACM.

Meij, E., K. Balog, and D. Odijk (2014). "Entity Linking and Retrieval for Semantic Search". In: *WSDM '14*. ACM.

Meij, E., M. Bron, L. Hollink, B. Huurnink, and M. de Rijke (2011). "Mapping Queries to the Linking Open Data Cloud: A Case Study Using DBpedia". *Web Semantics: Science, Services and Agents on the World Wide Web*. 9(4): 418–433.

Meij, E., M. Bron, B. Huurnink, L. Hollink, and M. de Rijke (2009). "Learning Semantic Query Suggestions". In: *ISWC '09*. Springer-Verlag.

Meij, E., W. Weerkamp, and M. de Rijke (2012). "Adding Semantics to Microblog Posts". In: *WSDM '12*. ACM.
Melo, G. de and N. Tandon (2016). "Seeing is Believing: The Quest for Multimodal Knowledge". *SIGWEB Newsletter*. (Spring): 4:1–4:9.
Metzler, D. and W. B. Croft (2005). "A Markov Random Field Model for Term Dependencies". In: *SIGIR '05*. ACM.
Metzler, D. and W. B. Croft (2007). "Latent Concept Expansion Using Markov Random Fields". In: *SIGIR '07*. ACM.
Mihalcea, R. and A. Csomai (2007). "Wikify!: Linking Documents to Encyclopedic Knowledge". In: *CIKM '07*. ACM.
Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *EMNLP '04*. ACL.
Miliaraki, I. and R. Blanco (2015). "From Selena Gomez to Marlon Brando: Understanding Explorative Entity Search". In: *WWW '15*. ACM.
Milne, D. and I. H. Witten (2008). "Learning to Link with Wikipedia". In: *CIKM '08*. ACM.
Minard, A.-L., M. Speranza, R. Urizar, B. Altuna, M. van Erp, A. Schoen, and C. van Son (2016). "MEANTIME, the NewsReader Multilingual Event and Time Corpus". In: *LREC '16*. ELRA.
Mintz, M., S. Bills, R. Snow, and D. Jurafsky (2009). "Distant Supervision for Relation Extraction Without Labeled Data". In: *ACL '09*. ACL.
Mitra, B. and N. Craswell (2018). "An Introduction to Neural Information Retrieval". *Foundations and Trends in Information Retrieval*. 13(1): 1–126.
Mohapatra, H., S. Jain, and S. Chakrabarti (2013). "Joint Bootstrapping of Corpus Annotations and Entity Types". In: *EMNLP '13*. ACL.
Nakashole, N., T. Tylenda, and G. Weikum (2013). "Fine-grained Semantic Typing of Emerging Entities". In: *ACL '13*. ACL.
Navalpakkam, V., L. Jentzsch, R. Sayres, S. Ravi, A. Ahmed, and A. Smola (2013). "Measurement and Modeling of Eye-mouse Behavior in the Presence of Nonlinear Page Layouts". In: *WWW '13*. ACM.
Nickel, M., K. Murphy, V. Tresp, and E. Gabrilovich (2016). "A Review of Relational Machine Learning for Knowledge Graphs". *Proceedings of the IEEE*. 104(1): 11–33.
Nickel, M., V. Tresp, and H.-P. Kriegel (2011). "A Three-Way Model for Collective Learning on Multi-Relational Data". In: *ICML '11*. Omnipress.
Nickel, M., V. Tresp, and H.-P. Kriegel (2012). "Factorizing YAGO: Scalable Machine Learning for Linked Data". In: *WWW '12*. ACM.
Nikolaev, F., A. Kotov, and N. Zhiltsov (2016). "Parameterized Fielded Term Dependence Models for Ad-hoc Entity Retrieval from Knowledge Graph". In: *SIGIR '16*. ACM.

Odijk, D., E. Meij, and M. de Rijke (2013). "Feeding the Second Screen: Semantic Linking Based on Subtitles". In: *OAIR '13*.

Odijk, D., E. Meij, I. Sijaranamual, and M. de Rijke (2015). "Dynamic Query Modeling for Related Content Finding". In: *SIGIR '15*. ACM.

Ojha, P. and P. P. Talukdar (2017). "KGEval - Estimating Accuracy of Automatically Constructed Knowledge Graphs". In: *EMNLP '17*. ACL.

Onal, K. D., Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan, T. McDonnell, A. T. Nguyen, D. Xu, B. C. Wallace, M. de Rijke, and M. Lease (2018). "Neural Information Retrieval: At the End of the Early Years". *Information Retrieval Journal*. 21(2–3): 111–182.

Oosterhuis, H. and M. de Rijke (2018). "Ranking for Relevance and Display Preferences in Complex Presentation Layouts". In: *SIGIR '18*. ACM.

Pantel, P. and A. Fuxman (2011). "Jigs and Lures: Associating Web Queries with Structured Entities". In: *ACL-HLT '11*. ACL.

Pasca, M., D. Lin, J. Bigham, A. Lifchits, and A. Jain (2006). "Organizing and Searching the World Wide Web of Facts - Step One: The One-million Fact Extraction Challenge". In: *AAAI'06*. AAAI Press.

Passos, A., V. Kumar, and A. McCallum (2014). "Lexicon Infused Phrase Embeddings for Named Entity Resolution". In: *CoNLL '14*. ACL.

Petkova, D. and W. B. Croft (2008). "Hierarchical Language Models for Expert Finding in Enterprise Corpora". *International Journal on Artificial Intelligence Tools*. 17: 5–18.

Potthast, M., B. Stein, and R. Gerling (2008). "Automatic Vandalism Detection in Wikipedia". In: *ECIR '08*. Springer-Verlag.

Pound, J., A. K. Hudek, I. F. Ilyas, and G. Weddell (2012). "Interpreting Keyword Queries over Web Knowledge Bases". In: *CIKM '12*. ACM.

Pound, J., P. Mika, and H. Zaragoza (2010). "Ad-hoc Object Retrieval in the Web of Data". In: *WWW '10*. ACM.

Prokofyev, R., G. Demartini, and P. Cudré-Mauroux (2014). "Effective Named Entity Recognition for Idiosyncratic Web Collections". In: *WWW '14*. ACM.

Pujara, J., E. Augustine, and L. Getoor (2017). "Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short". In: *EMNLP '15*. ACL.

Ran, C. and J. Wang (2018). "An Attention Factor Graph Model for Tweet Entity Linking". In: *WWW '18*. ACM.

Ratinov, L., D. Roth, D. Downey, and M. Anderson (2011). "Local and Global Algorithms for Disambiguation to Wikipedia". In: *ACL-HLT '11*. ACL.

Raviv, H., O. Kurland, and D. Carmel (2016). "Document Retrieval Using Entity-Based Language Models". In: *SIGIR '16*. ACM.

Reinanda, R., E. Meij, and M. de Rijke (2015). "Mining, Ranking and Recommending Entity Aspects". In: *SIGIR '15*. ACM.

Reinanda, R., E. Meij, and M. de Rijke (2016). "Document Filtering for Long-tail Entities". In: *CIKM '16*. ACM.

Ren, X., W. He, M. Qu, L. Huang, H. Ji, and J. Han (2016a). "AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-label Embedding". In: *EMNLP '16*. *ACL*.

Ren, X., W. He, M. Qu, C. R. Voss, H. Ji, and J. Han (2016b). "Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding". In: *KDD '16*. ACM.

Ren, X., A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han (2015). "ClusType: Effective Entity Recognition and Typing by Relation Phrase-Based Clustering". In: *KDD '15*. ACM.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *KDD '16*. ACM.

Riedel, S., L. Yao, B. M. Marlin, and A. McCallum (2013). "Relation Extraction with Matrix Factorization and Universal Schemas". In: *HLT-NAACL '13*. ACL.

Riedel, S., L. Yao, and A. McCallum (2010). "Modeling Relations and Their Mentions Without Labeled Text". In: *ECML PKDD '10*. Springer-Verlag.

Riloff, E. and R. Jones (1999). "Learning Dictionaries for Information Extraction by Multi-level Bootstrapping". In: *AAAI '99*. AAAI Press.

Sanderson, M. (2015). "Test Collection Based Evaluation of Information Retrieval Systems". *Foundations and Trends in Information Retrieval*. 4(4): 247–375.

Sarawagi, S. and W. W. Cohen (2004). "Semi-Markov Conditional Random Fields for Information Extraction". In: *NIPS '04*. JLMR.

Sarmento, L., V. Jijkoun, M. de Rijke, and E. Oliviera (2007). ""More Like These": Growing Entity Classes from Seeds". In: *CIKM '07*. ACM.

Sarrafzadeh, B., O. Vechtomova, and V. Jokic (2014). "Exploring Knowledge Graphs for Exploratory Search". In: *IIiX '14*. ACM.

Schuhmacher, M., L. Dietz, and S. Paolo Ponzetto (2015). "Ranking Entities for Web Queries Through Text and Knowledge". In: *CIKM '15*. ACM.

Sekine, S. (1998). "NYU: Description of the Japanese NE system used for MET-2". In: *MUC-7*. *ACL*.

Sekine, S. (2009). *Named Entities: Recognition, classification and use*. John Benjamin Publishings.

Sekine, S. and C. Nobata (2004). "Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy". In: *LREC*. ACL.

Seufert, S., K. Berberich, S. J. Bedathur, S. K. Kondreddi, P. Ernst, and G. Weikum (2016). "ESPRESSO: Relationships Between Entity Sets". In: *CIKM '16*. ACM.

Sevgili, Ö., A. Panchenko, and C. Biemann (2019). "Improving Neural Entity Disambiguation with Graph Embeddings". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Florence, Italy: Association for Computational Linguistics. 315–322. DOI: 10.18653/v1/P19-2044. URL: https://www.aclweb.org/anthology/P19-2044.

Shen, J., J. Xiao, X. He, J. Shang, S. Sinha, and J. Han (2018). "Entity Set Search of Scientific Literature: An Unsupervised Ranking Approach". In: *SIGIR '18*.

Shimaoka, S., P. Stenetorp, K. Inui, and S. Riedel (2016). "An Attentive Neural Architecture for Fine-grained Entity Type Classification". In: *AKBC '16*. ACL.

Shimaoka, S., P. Stenetorp, K. Inui, and S. Riedel (2017). "Neural Architectures for Fine-grained Entity Type Classification". In: *EACL '17*. ACL.

Shirakawa, M., K. Nakayama, T. Hara, and S. Nishio (2013). "Probabilistic Semantic Similarity Measurements for Noisy Short Texts Using Wikipedia Entities". In: *CIKM '13*. ACM.

Shokouhi, M. and Q. Guo (2015). "From Queries to Cards: Re-ranking Proactive Card Recommendations Based on Reactive Search History". In: *SIGIR '15*. ACM.

Sil, A. and A. Yates (2013). "Re-ranking for Joint Named-Entity Recognition and Linking". In: *CIKM '13*. ACM.

Singh, J., J. Hoffart, and A. Anand (2016). "Discovering Entities with Just a Little Help from You". In: *CIKM '16*. ACM.

Singh, S., A. Subramanya, F. Pereira, and A. McCallum (2012). "Wikilinks: A Large-scale Cross-Document Coreference Corpus Labeled via Links to Wikipedia". *Tech. rep.* No. UM-CS-2012-015. University of Massachusetts, Amherst.

Socher, R., D. Chen, C. D. Manning, and A. Y. Ng (2013). "Reasoning with Neural Tensor Networks for Knowledge Base Completion". In: *NIPS '13*. Curran Associates Inc.

Socher, R., B. Huval, C. D. Manning, and A. Y. Ng (2012). "Semantic Compositionality Through Recursive Matrix-vector Spaces". In: *CoNLL '12*. *CoNLL '12*. ACL.

Strubell, E., P. Verga, D. Belanger, and A. McCallum (2017). "Fast and Accurate Entity Recognition with Iterated Dilated Convolutions". In: *ACL '17*. ACL.

Suchanek, F. M., G. Kasneci, and G. Weikum (2007). "Yago: A Core of Semantic Knowledge". In: *WWW '07*. ACM.

Sun, H., H. Ma, W.-t. Yih, C.-T. Tsai, J. Liu, and M.-W. Chang (2015). "Open Domain Question Answering via Semantic Enrichment". In: *WWW '15*. ACM.

Surdeanu, M., S. Gupta, J. Bauer, D. McClosky, A. X. Chang, V. I. Spitkovsky, and C. D. Manning (2011). "Stanford's Distantly-Supervised Slot-Filling System". In: *Proceedings of the TAC-KBP 2011 Workshop*. NIST.

Surdeanu, M., J. Tibshirani, R. Nallapati, and C. D. Manning (2012). "Multi-instance Multi-label Learning for Relation Extraction". In: *CoNLL '12*. ACL.

Sutskever, I., J. B. Tenenbaum, and R. R. Salakhutdinov (2009). "Modelling Relational Data using Bayesian Clustered Tensor Factorization". In: *Advances in Neural Information Processing Systems*. NIPS '09. 1821–1828.

Tan, C. H., E. Agichtein, P. Ipeirotis, and E. Gabrilovich (2014). "Trust, but verify: Predicting Contribution Quality for Knowledge Base Construction and Curation". In: *WSDM '14*. ACM.

Tang, J., Z. Fang, and J. Sun (2015). "Incorporating Social Context and Domain Knowledge for Entity Recognition". In: *WWW '15*. ACM.

Tao, F., B. Zhao, A. Fuxman, Y. Li, and J. Han (2015). "Leveraging Pattern Semantics for Extracting Entities in Enterprises". In: *WWW '15*. ACM.

Tjong Kim Sang, E. F. and F. De Meulder (2003). "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *CoNLL '03*. ACL.

Tong, H. and C. Faloutsos (2006). "Center-piece Subgraphs: Problem Definition and Fast Solutions". In: *KDD '06*. ACM.

Tonon, A., G. Demartini, and P. Cudré-Mauroux (2012). "Combining Inverted Indices and Structured Search for Ad-hoc Object Retrieval". In: *SIGIR '12*. ACL.

Toutanova, K. and D. Chen (2015). "Observed Versus Latent Features for Knowledge Base and Text Inference". In: *3rd Workshop on Continuous Vector Space Models and Their Compositionality*. ACL.

Toutanova, K., D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon (2015). "Representing Text for Joint Embedding of Text and Knowledge Bases". In: *EMNLP '15*. ACL.

Usbeck, R., M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, and L. Wesemann (2015). "GERBIL: General Entity Annotator Benchmarking Framework". In: *WWW '15*. ACM.

Van Gysel, C., E. Kanoulas, and M. de Rijke (2018). "A Neural Vector Space Model". *ACM Transactions on Information Systems*. 36(4): 38.

Van Gysel, C., M. de Rijke, and E. Kanoulas (2016a). "Learning Latent Vector Spaces for Product Search". In: *CIKM '16*. ACM.

Van Gysel, C., M. de Rijke, and E. Kanoulas (2017a). "Semantic Entity Retrieval Toolkit". In: *SIGIR 2017 Workshop on Neural Information Retrieval)*. ACM.

Van Gysel, C., M. de Rijke, and E. Kanoulas (2017b). "Structural Regularities in Text-based Entity Vector Spaces". In: *ICTIR '17*. ACM.

Van Gysel, C., M. de Rijke, and M. Worring (2016b). "Unsupervised, Efficient and Semantic Expertise Retrieval". In: *WWW '16*. ACM.

Vannella, D., D. Jurgens, D. Scarfini, D. Toscani, and R. Navigli (2014). "Validating and Extending Semantic Knowledge Bases using Video Games with a Purpose". In: *ACL '14*. ACL.

Voorhees, E. M. (2005). "The TREC Robust Retrieval Track". *SIGIR Forum*. 39(1): 11–20.

Voskarides, N., E. Meij, and M. de Rijke (2017). "Generating Descriptions of Entity Relationships". In: *ECIR '17*. Springer-Verlag.

Voskarides, N., E. Meij, M. Tsagkias, M. de Rijke, and W. Weerkamp (2015). "Learning to Explain Entity Relationships in Knowledge Graphs". In: *ACL '15*. ACL.

Wang, J., D. Song, C. Lin, and L. Liao (2013a). "BIT and MSRA at TREC KBA CCR Track 2013". In: *TREC*. NIST.

Wang, J., D. Song, Q. Wang, Z. Zhang, L. Si, L. Liao, and C.-Y. Lin (2015a). "An Entity Class-Dependent Discriminative Mixture Model for Cumulative Citation Recommendation". In: *SIGIR '15*. ACM.

Wang, R. C. and W. W. Cohen (2007). "Language-Independent Set Expansion of Named Entities Using the Web". In: *ICDM '07*. IEEE.

Wang, T.-X., K.-Y. Tsai, and W.-H. Lu (2014a). "Identifying Real-Life Complex Task Names with Task-Intrinsic Entities from Microblogs". In: *ACL '14*. ACL.

Wang, X., D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua (2018). "Explainable Reasoning over Knowledge Graphs for Recommendation". In: *AAAI '18*. AAAI Press.

Wang, X., X. L. Dong, and A. Meliou (2015b). "Data X-Ray: A Diagnostic Tool for Data Errors". In: *SIGMOD '15*. ACM.

Wang, Z., J. Zhang, J. Feng, and Z. Chen (2014b). "Knowledge Graph Embedding by Translating on Hyperplanes". In: *AAAI '14*. AAAI Press.

Wang, Z., J. Li, Z. Wang, and J. Tang (2012). "Cross-lingual Knowledge Linking Across Wiki Knowledge Bases". In: *WWW '12*. ACM.

Wang, Z., Z. Li, J. Li, J. Tang, and J. Z. Pan (2013b). "Transfer Learning Based Cross-lingual Knowledge Extraction for Wikipedia". In: *ACL '13*. ACL.

West, R., E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin (2014). "Knowledge Base Completion via Search-based Question Answering". In: *WWW '14*. ACM.

Woods, W. A. (1977). "Lunar Rocks in Natural English: Explorations in Natural Language Question Answering". In: *Linguistic Structures Processing*. Ed. by A. Zampoli. Elsevier North-Holland. 521–569.

Wu, F. and D. S. Weld (2010). "Open Information Extraction Using Wikipedia". In: *ACL '10*. ACL.

Wu, Q., D. Teney, P. Wang, C. Shen, A. R. Dick, and A. van den Hengel (2016a). "Visual Question Answering: A Survey of Methods and Datasets". *arXiv preprint arXiv:1607.05910*.

Wu, Z., Y. Song, and C. L. Giles (2016b). "Exploring Multiple Feature Spaces for Novel Entity Discovery". In: *AAAI '16*. AAAI Press.

Xie, X., J. Mao, M. de Rijke, R. Zhang, M. Zhang, and S. Ma (2018). "Constructing an Interaction Behavior Model for Web Image Search". In: *SIGIR '18*. ACM.

Xiong, C. and J. Callan (2015a). "EsdRank: Connecting Query and Documents through External Semi-Structured Data". In: *CIKM '15*. ACM.

Xiong, C. and J. Callan (2015b). "Query Expansion with Freebase". In: *ICTIR '15*. ACM.

Xiong, C., J. Callan, and T.-Y. Liu (2017a). "Word-Entity Duet Representations for Document Ranking". In: *SIGIR '17*. ACM.

Xiong, C., R. Power, and J. Callan (2017b). "Explicit Semantic Ranking for Academic Search via Knowledge Graph Embedding". In: *WWW '17*. ACM.

Xu, P. and D. Barbosa (2018). "Neural Fine-Grained Entity Type Classification with Hierarchy-Aware Loss". In: *EMNLP '18*. ACL.

Yaghoobzadeh, Y. and H. Schutze (2015). "Corpus-level Fine-grained Entity Typing Using Contextual Information". In: *EMNLP '15*. ACL.

Yahya, M., D. Barbosa, K. Berberich, Q. Wang, and G. Weikum (2016). "Relationship Queries on Extended Knowledge Graphs". In: *WSDM '16*. ACM.

Yang, B., W. Yih, X. He, J. Gao, and L. Deng (2015). "Embedding Entities and Relations for Learning and Inference in Knowledge Bases". In: *ICLR '15*. ICLR.

Yao, L., S. Riedel, and A. McCallum (2010). "Collective Cross-document Relation Extraction Without Labelled Data". In: *EMNLP '10*. ACL.

Yao, X. and B. Van Durme (2014). "Information Extraction over Structured Data: Question Answering with Freebase". In: *ACL '14*. ACL.

Yih, W.-T., M.-W. Chang, X. He, and J. Gao (2015). "Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base". In: *ACL '15*. ACL.

Ying, R., R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec (2018). "Graph convolutional neural networks for web-scale recommender systems". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 974–983.

Yogatama, D., D. Gillick, and N. Lazic (2015). "Embedding Methods for Fine Grained Entity Type Classification". In: *ACL '15*. ACL.

Yosef, M. A., S. Bauer, J. Hoffart, M. Spaniol, and G. Weikum (2012). "HYENA: Hierarchical Type Classification for Entity Names". In: *COLING '12*. ACL.

Yu, X., H. Ma, B.-j. P. Hsu, and J. Han (2014a). "On Building Entity Recommender Systems Using User Click Log and Freebase Knowledge". In: *WSDM '14*. *ACM*.

Yu, X., X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han (2014b). "Personalized Entity Recommendation: A Heterogeneous Information Network Approach". In: *WSDM '14*. ACM.

Zelenko, D., C. Aone, and A. Richardella (2002). "Kernel Methods for Relation Extraction". In: *EMNLP '02*. *ACL*.

Zeng, D., K. Liu, Y. Chen, and J. Zhao (2015). "Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks". In: *EMNLP '15*. ACL.

Zhang, C., G. Zhou, Q. Lu, and F. Chang (2017a). "Graph-based Knowledge Reuse for Supporting Knowledge-driven Decision-making in New Product Development". *International Journal of Production Research*. 55(23): 7187–7203.

Zhang, F., N. J. Yuan, D. Lian, X. Xie, and W. Y. Ma (2016a). "Collaborative Knowledge Base Embedding for Recommender Systems". In: *KDD '16*. *ACM*.

Zhang, L., M. Färber, and A. Rettinger (2016b). "XKnowSearch!: Exploiting Knowledge Bases for Entity-based Cross-lingual Information Retrieval". In: *CIKM '16*. ACM.

Zhang, L., A. Rettinger, and J. Zhang (2016c). "A Knowledge Base Approach to Cross-Lingual Keyword Query Interpretation". In: *ISWC '16*. Springer-Verlag.

Zhang, W., B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, and H. Chen (2019). "Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning". In: *WWW '19*. ACL.

Zhang, Y., T. Paradis, L. Hou, J. Li, J. Zhang, and H. Zheng (2017b). "Cross-Lingual Infobox Alignment in Wikipedia Using Entity-Attribute Factor Graph". In: *ISWC '17*. Springer-Verlag.

Zhao, S. and Y. Zhang (2014). "Tailor Knowledge Graph for Query Understanding: Linking Intent Topics by Propagation". In: *EMNLP '14*. ACL.

Zhao, S. and R. Grishman (2005). "Extracting Relations with Integrated Information Using Kernel Methods". In: *ACL '05*. ACL.

Zhiltsov, N. and E. Agichtein (2013). "Improving Entity Search over Linked Data by Modeling Latent Semantics". In: *CIKM '13*. ACM.

Zhiltsov, N., A. Kotov, and F. Nikolaev (2015). "Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data". In: *SIGIR '15*. ACM.

Zhong, H., J. Zhang, Z. Wang, H. Wan, and Z. Chen (2015). "Aligning Knowledge and Text Embeddings by Entity Descriptions". In: *EMNLP '15*. ACL.

Zhou, M. and K. C.-C. Chang (2013). "Entity-centric Document Filtering: Boosting Feature Mapping Through Meta-features". In: *CIKM '13*. ACM.

Zhu, G. and C. A. Iglesias (2018). "Exploiting semantic similarity for named entity disambiguation in knowledge graphs". *Expert Systems with Applications*. 101(July): 8–24.

Zhu, Y., C. Zhang, C. Ré, and F.-f. Li (2015). "Building a Large-scale Multimodal Knowledge Base for Visual Question Answering". *arXiv preprint arXiv:1507.05670*.