Chapter 8
Entity Linking in Enterprise Search: Combining Textual and Structural Information

Sumit Bhatia

Abstract Fast and correct identification of named entities in queries is crucial for query understanding and to map the query to information in structured knowledge base. Most of the existing works have focused on utilizing search logs and manually curated knowledge bases for entity linking and often involve complex graph operations and are generally slow. We describe a simple, yet fast and accurate, probabilistic entity linking algorithm that can be used in enterprise settings where automatically constructed, domain-specific knowledge graphs are used. In addition to the linked graph structure, textual evidence from the domain-specific corpus is also utilized to improve the performance.

8.1 Introduction

With increasing popularity of virtual assistants like SIRI and Google Now, users are interacting with search systems by asking natural language questions that often contain named entity mentions. A large-scale study by Pang and Kumar [40] observed statistically significant temporal increases in the fraction of questions–queries received by search engines and searchers tend to use more question–queries for complex information needs [3]. In case of web search engines, a large fraction of queries contain a named entity (estimates vary from 40% [31] to 60% [42]). Hence, *fast* and *correct* identification of named entities in user queries is crucial for query understanding and to map the query to information in structured knowledge base. Advancements in semantic search technology have enabled modern information retrieval systems to utilize structured knowledge bases such as DBPedia [2] and Yago [45] to satisfy users' information needs.

Most of the existing works on entity linking focus on linking the entities in long documents [26, 30]. These methods make use of the large context around the target

S. Bhatia (✉)
IBM Research AI, New Delhi, India
e-mail: sumitbhatia@in.ibm.com

Deepak P, A. Jurek-Loughrey (eds.), *Linking and Mining Heterogeneous and Multi-view Data*, Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-01872-6_8

mention in the document. Therefore, these methods are limited to perform on long text documents. However, some methods have been proposed that perform entity linking in short sentences [20, 27]. They rely on the collective disambiguation [15] of all the entity mentions appear in the sentences. Thus, these methods take long time in computing the confidence scores for all the combinations.

Most of the existing work on entity linking in search queries utilizes information derived from query logs and open knowledge bases such as DBPedia and Freebase (Sect. 8.2). Such techniques, however, are not suited for enterprise and domain-specific search systems such as legal, medical, and healthcare, due to very small user bases resulting in small query logs and the absence of rich domain-specific knowledge bases. Recently, there have been development of systems for automatic construction of semantic knowledge bases for domain-specific corpora [12, 48] and systems that use such domain-specific knowledge bases [38]. In this chapter, we describe a method for entity disambiguation and linking, developed especially for enterprise settings, where such external resources are often not available. The proposed system offers users a search interface to search for the indexed information and uses the underlying knowledge base to enhance search results and provide additional entity-centric data exploration capabilities that allow users to explore hidden relationships between entities discovered automatically from a domain-specific corpus.

The system *automatically* constructs a structured knowledge base by identifying entities and their relationships from input text corpora using the method described by Castelli et al. [12]. Thus, for each relationship discovered by the system, the corresponding mention text provides additional contextual information about the entities and relationships present in that mention. We posit that the *dense graph structure* discovered from the corpus, as well as the *additional context provided by the associated mention text*, can be utilized together for linking entity name mentions in search queries to corresponding entities in the graph. Our proposed entity linking algorithm is intuitive, relies on a theoretical sound probabilistic framework, and is fast and scalable with an average response time of $\approx$ **87** ms. Figure 8.1 shows the working of proposed algorithm in action where top ranked

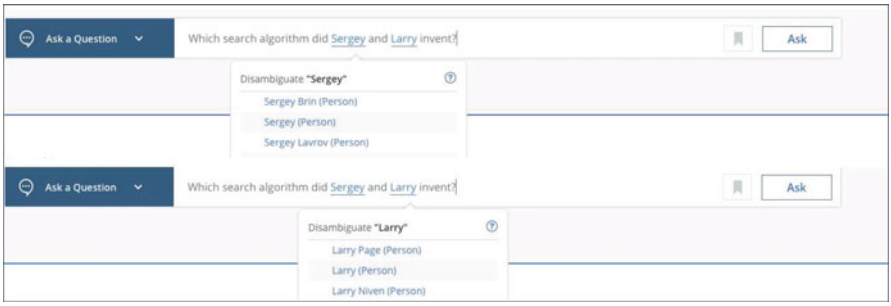

Fig. 8.1 Entity suggestions produced by proposed approach using text and entity context in search query

suggestions for named mentions `Sergey` and `Larry` are showed. As will be described in detail in Sect. 8.3, note that the algorithm is making these suggestions by utilizing the terms in questions (search, algorithm) as well as relationships between all target entities for mentions "Sergey" and "Larry" in the graph. The algorithm figures out that entities "Sergey Brin" and "Larry Page" have strong evidences from their textual content as well as these two entities are strongly connected in the graph, and hence they are suggested as most probable relevant entities in the context of question.

The material presented in this chapter is an extended version of our ESWC 2016 paper [5], and we provide a detailed survey of the representative work on entity linking and discuss their shortcomings when applied to enterprise settings. We describe our proposed approach in detail with several examples to illustrate the working of the algorithm. We hope that the additional details will help the readers, especially beginners and practitioners, to understand the finer details and workings of the proposed approach and will help them implement the approach for their custom applications.

8.2 Related Work

We first discuss early works that provide the foundation for the general entity linking task and define the problem in context of knowledge graphs. We then review representative works that addressed entity linking in longer documents as well as much shorter text fragments such as web queries and tweets.

8.2.1 Entity Linking Background

At its core, the problem of *entity linking* is similar to the general problem of *record linkage* that was first introduced by Dunn [18] in the context of assembling all public records of an individual. This idea was further popularized by Newcombe et al. [39] that proposed the use of computers to link multiple separately recorded pieces of information about a particular person or family for census applications. In general, *record linkage* refers to the task of finding and linking records about an entity spread across multiple datasets. This is an extensively studied problem in the field of databases and data mining, and a detailed survey is out of the scope of this chapter. We direct the interested reader to excellent surveys on this topic by Brizan and Tansel [11] and Christen [14].

Entity linking, as studied in this chapter, refers to the task of linking the mention of a named entity in text (a sentence, keyword query, etc.) to the corresponding entity in a knowledge base. Let us consider the following piece of text about Barack Obama to understand the challenges involved.

Barack Obama served as the 44^{th} President of the United States from 2009 to 2017. *He* was born in Honolulu, Hawaii. *Obama* has two daughters.

A named entity recognizer [35, 37] when run on the above text will be able to identify *Barack Obama* and *Obama* as two named entities. However, these two different *surface forms* correspond to the same entity $BarackObama$ in the underlying knowledge base. Hence, it is required for the system to be able to identify that these two different mentions are variations of the same entity name and link them to the same canonical entity, a task known as *entity normalization* [29]. Also, note that in the above example text, the pronoun *he* also refers to Barack Obama. This task of determining different expressions (nouns, pronouns, etc.) that refer to the same entity is known as *coreference resolution* [19]. Note that depending upon the requirements, it may also be required to perform coreference resolution and entity normalization across multiple documents [4, 28, 41]. While the tasks of named entity recognition, coreference resolution, and entity normalization have been studied extensively, entity linking involves the additional step of aligning the identified and normalized entity mention to its corresponding entity in the knowledge base.

8.2.2 Linking Entities in Documents and Web Pages

Entity linking has been studied under various application scenarios. SemTag [17] was one of the first systems to consider the task of linking entities in web pages to entities in a knowledge base (Stanford TAP entity catalog [22]). Wikipedia, owing to exhaustive coverage of general concepts, has been used as the underlying knowledge base to link entity mentions in documents, web pages, news articles, etc. [15, 26, 34, 36, 43]. Mihalcea and Csomai [34] introduced the *Wikify!* system to extract keywords from documents and link them to their corresponding Wikipedia pages. Cucerzan [15] utilized Wikipedia category information, in addition to contextual similarities between documents and Wikipedia based features entity normalization and linking. Kulkarni et al. [30] premised that entities mentioned in a coherent document are topically related and showed that utilizing this information to collectively link entities in a document can help improve performance. Hoffart et al. [26] proposed a comprehensive framework for collective entity disambiguation and linking that combines local contextual information about the input mention with coherence among candidate entities for all entity mentions together.

8.2.3 Linking Entities in Short Text Fragments

The methods discussed till now have focused on performing entity linking for longer documents like web pages, news articles, etc. Such documents generally contain enough contextual clues as well as additional metadata that could aid identifying

appropriate mentions. In case of shorter text documents, such as microblogs, or web search queries that are generally a few keywords long, successful entity linking has to rely on specific application specific contextual clues and metadata in absence of large document context. For example, in case of linking entity mentions in tweets, user characteristics, interest profiles, social network properties such as retweets and likes can be utilized [23, 32, 44]. Ferragina and Scaiella [20] utilize the anchor texts of Wikipedia articles to construct a dictionary of different surface forms or name variations for entities and use that to identify entity mentions in short text fragments. The final set of target entities is then determined by collective agreement among different potential mappings. Hoffart et al. [27] describe an algorithm that performs collective entity linking by computing overlap between the sets of keywords associated with each target entity. For creating the set of keywords, noun phrases from Wikipedia entity pages are used. The proposed algorithm achieves good performance for both short and long texts, as well as for less popular, long tail entities.

Another challenging setting for performing entity linking is in the context of web search queries that are often just a collection of few keywords. Typical ways to perform entity linking in such systems is to approximate semantic similarity between queries and entities by utilizing their respective language models [21, 25]. Successful identification and linking of entity mentions in queries can also help improve retrieval performance by means of query expansion using entity features from the knowledge base [16]. Another challenge for entity linking in search systems is that it has to be performed before the actual retrieval takes place and thus, needs to be completed in just a few milliseconds. Blanco, Ottaviano, and Meij [10] describe a space efficient algorithm for entity linking for web search queries that is able to process queries in sub-milliseconds time.

These methods use features derived from query logs to gather user context, target documents, etc., to get context. However, in many enterprise systems, such additional metadata is not readily available [7]. Further, the knowledge bases used in such systems may not be as rich as Wikipedia lacking hyperlinks, metadata, etc., and are often constructed using automated methods [8]. However, context is important [6]. In this work, we discuss how we can utilize the limited context available in the input query (text, entity mentions) and utilize the textual information in background corpus coupled with rich graph structure to perform entity linking in enterprise search systems.

8.3 Proposed Approach

We first describe the problem setting and our assumptions, and provide a probabilistic formulation of the entity linking problem. We also discuss how different application settings can be mapped to the proposed formulation and then provide a solution for entity linking that utilizes structural properties of entities in the knowledge graph and information from the background text corpus.

8.3.1 Problem Setting

Let us consider a knowledge graph $\mathcal{K} = \{\mathcal{E}, \mathcal{R}\}$ where $\mathcal{E}$ is the set of entities (nodes) and $\mathcal{R}$ is the set of relationships (edges). Let us also assume the availability of a background text corpus $\mathcal{C}$.[1] Let $\mathcal{M}_r$ be the set of all the mentions of the relationship r in the background text corpus. As an example, consider the relationship $\langle SteveJobs \rangle$, $founderOf$, $\langle AppleInc. \rangle$ and one of its many mentions from Wikipedia, *"Jobs and Wozniak co-founded Apple in 1976 to sell Wozniak's Apple I personal computer."* Note that in addition to the relationship under consideration, this mention also provides additional contextual clues about the entities $SteveJobs$ and $AppleInc$. (Wozniak, personal computer are related to Steve Jobs and Apple Inc.)

8.3.2 Problem Formulation

Let $Q = \{C, T\}$ be the input query where T is the ambiguous token, and $C = \{E_c, W_c\}$ is the context under which we have to disambiguate T. The context is provided by the words ($W_c = \{w_{c1}, w_{c2}, \ldots, w_{cl}\}$) in the query and the set of unambiguous entities $E_c = \{e_{c1}, e_{c2}, \ldots, e_{cm}\}$. Note that initially, this entity set can be empty if there are no unambiguous entity mentions in the query and in such cases, only textual information is considered. The task is to map the ambiguous token T to one of the possible target entities.

This is a generalized statement of the entity linking task and covers a variety of end-applications and scenarios as discussed below.

- **Search Systems:** The user typically enters a few keywords and the task is to link the keywords in query to an entity in the knowledge graph. Note that not all the terms in the query correspond to entity mentions and the problem is further exacerbated by the inherent ambiguity of keyword queries [24]. For example, in the query `obama speeches`, `obama` corresponds to the entity *Barack Obama* and `speeches` provides the information need of the user. Also note that keyword queries lack the additional contextual information that is present while linking entities in documents. To overcome this, web search systems often utilize query logs and user activity to gather context about users' information needs [24]. Once terms in the queries are linked to corresponding entities in the graph, related entities can also be offered as recommendations to the end-user for further browsing [9].

[1] For domain-specific applications where the knowledge graph is constructed using automated methods, the set of input documents constitute the background corpus. For applications that use generic, open-domain knowledge bases such as DBPedia and WikiData, Wikipedia could be used as the background text corpus.

- **Question Answering Systems:** By identifying entities of interest in the question, the underlying knowledge base can be used to retrieve the appropriate facts required to answer the question [47]. In a typical QA system, the user enters a natural language question such as `When did Steve become ceo of Microsoft?` Here, the terms of interest are `Steve` and `Microsoft`. Also note that in this example, `Microsoft` also provides contextual evidence that provides additional support for *Steve Ballmer* compared to many other person entities named Steve such *Steve Jobs* or *Steve Wozniak* that will have less relevance to `Microsoft` than *Steve Ballmer*. Once the system correctly links `steve` to *Steve Ballmer*, appropriate facts from the knowledge graph can be easily retrieved and presented as answer to the user.

8.3.3 Proposed Solution

On receiving the input query, the first step in the solution to the problem as formulated above is to identify entity mentions in the query. These mentions are then linked to the corresponding entity in the knowledge graph. These entity mentions could be identified using NLP libraries such as Apache Open NLP[2] and Stanford Named Entity Recognizer.[3]

After identifying the token T that is a named entity mention in the query Q, the next step is to generate a list of target candidate entities. Such a list could be generated by using a dictionary that contains different surface forms of the entity names [30, 46, 49, 50]. For example, a dictionary could be constructed that maps different surface forms of the entity *Barack Obama* such as `Barack Obama, Barack H. Obama, and President Obama` to the entity. Since we are interested in mapping the token to entities in the knowledge graph $\mathcal{K} = \{\mathcal{E}, \mathcal{R}\}$, we select all the entities that contain token T as a sub-string in their name. For example, for the token `Steve` all entities such as `Steve Jobs, Steve Wozniak, and Steve Lawrence` constitute the set of target entities. Note that for domain-specific applications, such a dictionary could also be constructed by using domain-specific sources such as the gene and protein dictionaries used in the KnIT system for studying protein–protein interactions [38]. For generic, open-domain systems, Wikipedia has been used extensively to create such dictionaries by utilizing disambiguation and redirect pages, anchor text and hyperlinks, etc.

Formally, let $E_T = \{e_{T1}, e_{T2}, \ldots, e_{Tm}\}$ be the set of target entities for the ambiguous token T in the query. Using the context information, we can produce a ranked list of target entities by computing $P(e_{Ti}|C)$, i.e., the probability that the user is interested in entity e_{Ti} given the context C. Using Bayes' theorem, we can write $P(e_{Ti}|C)$ as follows:

[2] http://opennlp.apache.org/.

[3] https://nlp.stanford.edu/software/CRF-NER.html.

$$P(e_{Ti}|C) = \frac{P(e_{Ti})P(C|e_{Ti})}{P(C)} \tag{8.1}$$

Here $P(e_{Ti})$ represents the prior probability of the entity e_{Ti} to be relevant without any context information. This prior probability can be computed in multiple ways based on the application requirements. For example, priors can be computed based on frequency of individual entities or temporal information (such as recency) in case of news domain. In this work, we assume a frequency based prior indicating that in the absence of any context information, the probability of an entity being relevant is directly proportional to its frequency in the graph. Further, since we are only interested in relative ordering of the target entities, we can ignore the denominator $P(C)$ as its value will be same for all the target entities. With these assumptions, Eq. (8.1) can be re-written as follows:

$$P(e_{Ti}|C) \propto P(e_{Ti}) \times P(C|e_{Ti}) \tag{8.2}$$

Here $P(C|e_{Ti})$ represents the probability of observing the context C after having seen the entity e_{Ti}. Note that the context C consists of two components—text context and entity context. Assuming that the probability of observing text and entity context is conditionally independent, above equation can be reduced as follows:

$$P(e_{Ti}|C) \propto P(e_{Ti}) \times P(W_c|e_{Ti}) \times P(E_c|e_{Ti}) \tag{8.3}$$

$$= \underbrace{P(e_{Ti})}_{\text{entity prior}} \times \underbrace{\prod_{w_c \in W_c} P(w_c|e_{Ti})}_{\text{text context}} \times \underbrace{\prod_{e_c \in E_c} P(e_c|e_{Ti})}_{\text{entity context}} \tag{8.4}$$

8.3.3.1 Computing Entity Context Contribution

The *entity context* factor in Eq. (8.4) corresponds to the evidence for target entity given E_c, the set of entities forming the context. For each individual entity e_c forming the context, we need to compute $P(e_c|e_{Ti})$, i.e., the probability of observing e_c after observing the target entity e_{Ti}. Intuitively, there is a higher chance of observing an entity that is involved in multiple relationship with e_{Ti} than an entity that only has a few relationships with e_{Ti}. Thus, we can estimate $P(e_c|e_{Ti})$ as follows:

$$P(e_c|e_{Ti}) = \frac{\text{relCount}(e_c, e_{Ti}) + 1}{\text{relCount}(e_c) + |E|} \tag{8.5}$$

Note that the factor of 1 in numerator and $|E|$ (size of entity set E) in the denominator have been added to smoothen the probability values for entities that are not involved in any relationship with e_{Ti}.

8.3.3.2 Computing Text Context Contribution

The *text context* factor in Eq. (8.4) corresponds to the evidence for target entity given W_c, the terms present in the input query. For each individual query term w_c, we need to compute $P(w_c|e_{Ti})$, i.e., the probability of observing w_c given e_{Ti}. In order to compute this probability, we construct *mention language models* for each entity in the knowledge graph that capture different contexts in which the entity appears in the corpus.

To construct such a mention language model for an entity e, we need to capture all the *mention sentences*, i.e., the sentences from the text corpus that talk about entity e. In automatically constructed graphs, where rule based or machine learned systems identify entity and relationship mentions from text, the source text for each extracted relationship and entity can be utilized to capture all the mention sentences for entity e by combining all the source sentences from which the entity and its relationships were identified. The *mention documents* created in this way capture different contexts under which the entity has been observed in the input corpus. For example, a lot of relationships of Steve Jobs are with Apple products, executives, etc. So sentences for these relationships will contain mentions of things related to Apple, in addition to entity names. For example, sentences containing relationships of Steve Jobs with iPhone will contain words like *design, touchscreen, mobile, apps, battery,* etc., and all these *contextual clues* are captured in *mention document* for Steve Jobs.

The mention documents created in this way can be used to compute the probability $P(w_c|e_{Ti})$ as follows:

$$P(w_c|E_{Ti}) = P(w_c|M_{E_{Ti}}) \tag{8.6}$$

$$= \frac{\text{no. of times } w_c \text{ appears in } M_{E_{Ti}} + 1}{|M_{E_{Ti}}| + N} \tag{8.7}$$

Here N is the size of the vocabulary and $M_{E_{Ti}}$ is the mention document for entity E_{Ti}.

8.3.3.3 Putting It All Together

We now illustrate the working of the proposed approach through an example as illustrated in Fig. 8.2. Consider the input question, "Which search algorithm did sergey and larry invent." In this question, the NER module identifies *sergey* and *larry* as the two named entities that need to be linked to the corresponding entities in the knowledge graph. The two ambiguous tokens and the natural language question are fed as input to the system. As discussed, the first step is to generate a list of target entities that is performed by retrieving all entities from the graph containing *sergey* and *larry* in their names. For each such target entity, we need to compute

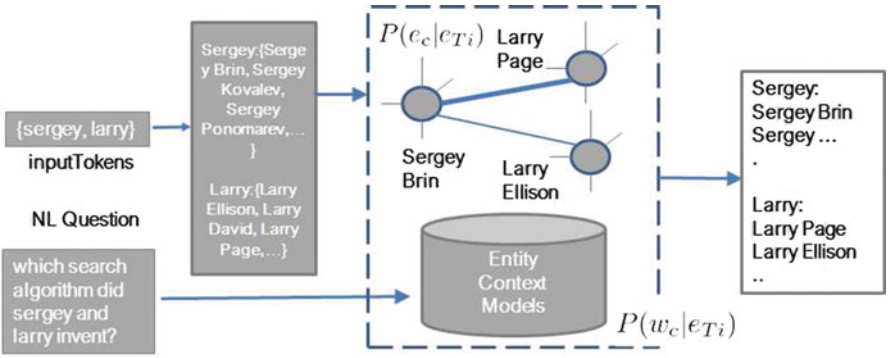

Fig. 8.2 Illustration of the proposed approach

the entity and text context components as described in Eq. (8.4). The entity context component helps in collective disambiguation of entities by taking into account the pairwise relevance of the target entities for the two ambiguous tokens. For example, the pair <Sergey Brin, Larry Page> will have much stronger connections in the graph (both Google co-founders share many common relations) compared to the pair <Sergey Brin, Larry Ellison> (Larry Ellison being co-founder of Oracle shares much less relations with Sergey Brin). Likewise, for the text context component, the mention language models of all target entities are used to find the entities that have the highest probability of generating the context terms in the questions such as *search and algorithm*. Thus, entities such as *Sergey Brin, Larry page, and Larry Ellison* get high text context component scores due to their relations with computer science related concepts. The two scores for all the target entities are combined to produce a final ranked list as illustrated in the figure.

8.4 Evaluation

8.4.1 Data Description

We use a semantic graph constructed from text of all articles in Wikipedia by automatically extracting the entities and their relations by using IBM's Watson natural language understanding (NLU) services.[4] Even though there exist popular knowledge bases like DBPedia that contain high quality data, we chose to construct a semantic graph using automated means as such a graph will be closer to many practical real-world scenarios where high quality curated graphs are often not available and one has to resort to automatic methods of constructing knowledge

[4] https://www.ibm.com/watson/services/natural-language-understanding/.

bases. Our graph contains more than 30 million entities and 192 million distinct relationships in comparison to 4.5 million entities and 70 million relationships in DBpedia.

8.4.2 Benchmark Test Set and Baselines

For evaluating the proposed approach, we use the KORE50 [27] dataset that contains 50 short sentences with highly ambiguous entity mentions (Table 8.1). This widely used dataset is considered among the hardest dataset for entity disambiguation and is being used widely for evaluating entity disambiguation/linking approaches. Further, on an average, there are only 14 words and roughly 3 mentions per sentence, thus making it ideal for evaluating our approach as it enables us to identify our interactive approach. Average sentence length (after stop word removal) is 6.88 words per sentence and each sentence has 2.96 entity mentions on an average. Every mention has an average of 631 candidates to disambiguate in YAGO knowledge base [45]. However, it varies for different knowledge bases. Our automatically constructed knowledge base has *2261 candidates per mention* to disambiguate illustrating the difficulty in entity linking due to high noise in automatically constructed knowledge bases when compared with manually curated/cleaned knowledge bases such as DBpedia. We also provide the performance numbers for a number of commonly used methods on the same dataset for reference [1, 13, 26, 27, 33] (Table 8.2).

Table 8.1 Characteristics of KORE50 dataset

Average sentence length	14.68
Average sentence length after stop word removal	6.88
Average entity mentions per sentence	2.96

Table 8.2 Entity disambiguation accuracy, measured in terms of precision, as achieved by the proposed approach

Method	Precision
Joint-DiSER-TopN [1]	0.72
AIDA-2012 [26]	0.57
AIDA-2013 [27]	0.64
Wikifier [13]	0.41
DBpedia spotlight [33]	0.35
Proposed method accuracy @ Rank 1	0.52
Proposed method accuracy @ Rank 5	0.65
Proposed method accuracy @ Rank 10	0.74

The table also provides accuracy achieved by several commonly used methods at Rank 1, as reported in the respective papers. For the proposed approach, precision achieved at Ranks 5 and 10 is also reported

8.4.3 Results and Discussions

The results of our proposed approach and various other state-of-the-art methods for entity linking on the same dataset are tabulated in Table 8.2. We note that on the standard KORE 50 dataset for entity disambiguation, our proposed approach, while being much simpler than the other reported methods, achieves comparable performance in terms of precision values at Rank 1. The top achieving methods do achieve better accuracy number but at the cost of higher complexity, reliance on many external resources of data, and consequently, slower speeds. For example, as reported by Hoffart et al. [27], the average time taken for disambiguation is 1.285 s with a standard deviation of 3.925 s. On the other hand, as we observe from Table 8.3, median response time for the proposed approach is about 86 ms, with the maximum response time being 125 ms. Such low response times were possible due to the fact that we utilized the signals from mention text and relationship information about entities that are much more computationally efficient to compute,[5] instead of performing complex and time-consuming graph operations as in other methods, while not sacrificing on the accuracy.

Figure 8.3 illustrates the working of proposed system in action for a variety of input <query,context> combinations. In Fig. 8.3a, the token `Steve` is provided without any context and the system returns a list sorted by entity prior (frequency). Next, in Fig. 8.3b–d, the results for the token `Steve` under different context terms are shown. Note how the system finds different entities in each case with changing context. Likewise, Fig. 8.3e shows the results for token `larry` without any context. However, as soon as we provide another token to disambiguate (`Sergey`) in Fig. 8.3f, the entity context component kicks in and collectively determines the most probable entities for both `sergey` and `larry`.

Table 8.3 Average candidate list size and response times per query

	Candidate size	Response time (ms)
Min.	0	85
Average	7917.27	87.34
Median	2261.5	86
Max.	183,546	125

The experiments were conducted on a standard MacBook Pro laptop with 16 GB RAM and an Intel i7 processor

[5] Text context components can be computed by using an inverted index implementation where using the context terms as queries, most relevant mention docs (and thus the corresponding entities) can be retrieved in a single query. Likewise, entity context component can be computed by just counting the number of connections between target entities—can be performed in a single optimized SQL query.

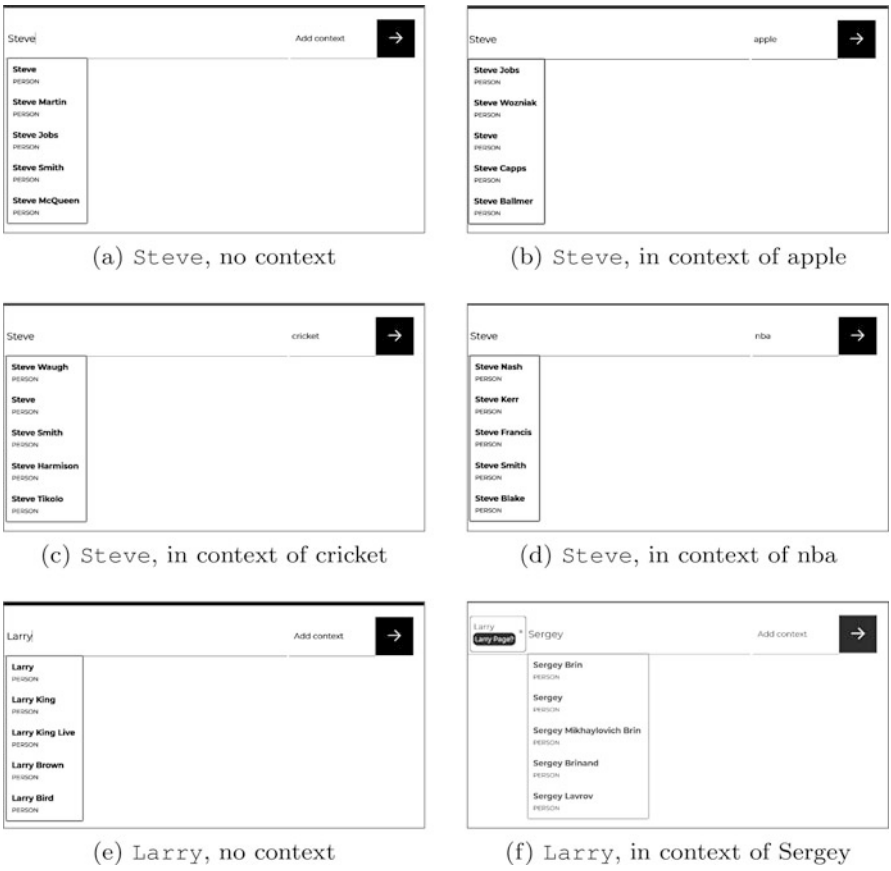

Fig. 8.3 Some examples of the proposed entity linking approach in action. Note how the suggestions for entities change in sub-figures (**a**)–(**d**) with varying contexts. Also note that how the entity context helps retrieve relevant results for `larry` in sub-figures (**e**) and (**f**). First, in sub-figure (**e**), in the absence of any context, the suggestions offered for `Larry` are simply ranked by the frequency prior, suggesting most popular entities containing `larry` in their name. Next, in sub-figure (**f**), when the user types `Sergey`, the system collectively disambiguates `Larry` as `Larry Page` and `Sergey` as `Sergey Brin`—note that this corresponds to the entity context component of the ranking function

8.5 Conclusions

In this chapter, we discussed the problem of mapping entity mentions in natural language search queries to corresponding entities in an automatically constructed knowledge graph. We provided a review of representative works on entity linking and their shortcomings when applied to enterprise settings. We then proposed an approach that utilizes the dense graph structure as well as additional context

provided by the mention text. Experimental evaluation on a standard dataset shows that the proposed approach is able to achieve high accuracy (comparable to other state-of-the-art methods) with a median response time of 86 ms.

References

1. Aggarwal, N., Buitelaar, P.: Wikipedia-based distributional semantics for entity relatedness. In: 2014 AAAI Fall Symposium Series (2014)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: Aberer, K., Choi, K.S., Noy, N.F., Allemang, D., Lee, K.I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) 6th International Semantic Web Conference (ISWC 2007). Lecture Notes in Computer Science, vol. 4825, pp. 722–735. Busan (2007). https://doi.org/10.1007/978-3-540-76298-0_52
3. Aula, A., Khan, R.M., Guan, Z.: How does search behavior change as search becomes more difficult? In: Mynatt, E.D., Schoner, D., Fitzpatrick, G., Hudson, S.E., Edwards, W.K., Rodden, T. (eds.) Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, GA, 10–15 April 2010, pp. 35–44. Association for Computing Machinery, New York (2010). http://doi.acm.org/10.1145/1753326.1753333
4. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: Boitet, C., Whitelock, P. (eds.) ACL/COLING, pp. 79–85. Morgan Kaufmann Publishers/ACL (1998). http://aclweb.org/anthology/P/P98/
5. Bhatia, S., Jain, A.: Context sensitive entity linking of search queries in enterprise knowledge graphs. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenic, D., Auer, S., Lange, C. (eds.) The Semantic Web – ESWC 2016 Satellite Events, Heraklion, Crete, 29 May–2 June 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9989, pp. 50–54 (2016). https://doi.org/10.1007/978-3-319-47602-5_11
6. Bhatia, S., Vishwakarma, H.: Know Thy Neighbors, and More! Studying the Role of Context in Entity Recommendation. In: HT '18: 29th ACM Conference on Hypertext and Social Media, 9–12 July 2018, Baltimore. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3209542.3209548
7. Bhatia, S., Majumdar, D., Mitra, P.: Query suggestions in the absence of query logs. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, pp. 795–804. Association for Computing Machinery, New York (2011). http://doi.acm.org/10.1145/2009916.2010023
8. Bhatia, S., Rajshree, N., Jain, A., Aggarwal, N.: Tools and infrastructure for supporting enterprise knowledge graphs. In: Cong, G., Peng, W., Zhang, W.E., Li, C., Sun, A. (eds.) Proceedings of the 13th International Conference Advanced Data Mining and Applications, ADMA 2017, Singapore, 5–6 November 2017. Lecture Notes in Computer Science, vol. 10604, pp. 846–852. Springer, Berlin (2017). https://doi.org/10.1007/978-3-319-69179-4_60
9. Blanco, R., Cambazoglu, B.B., Mika, P., Torzec, N.: Entity recommendations in web search. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) The Semantic Web – ISWC 2013, pp. 33–48. Springer, Berlin (2013)
10. Blanco, R., Ottaviano, G., Meij, E.: Fast and space-efficient entity linking for queries. In: Cheng, X., Li, H., Gabrilovich, E., Tang, J. (eds.) Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM 2015, Shanghai, 2–6 February 2015, pp. 179–188. Association for Computing Machinery, New York (2015). http://dl.acm.org/citation.cfm?id=2684822

11. Brizan, D.G., Tansel, A.U.: A. survey of entity resolution and record linkage methodologies. Commun. IIMA **6**(3), 5 (2006)
12. Castelli, V., Raghavan, H., Florian, R., Han, D.J., Luo, X., Roukos, S.: Distilling and exploring nuggets from a corpus. In: SIGIR, pp. 1006–1006 (2012)
13. Cheng, X., Roth, D.: Relational inference for wikification. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1787–1796. Association for Computational Linguistics, Seattle (2013). http://aclweb.org/anthology/D/D13/D13-1184.pdf
14. Christen, P.: Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Data-Centric Systems and Applications. Springer, Berlin (2012)
15. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Eisner, J. (ed.) EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 28–30 June 2007, Prague, pp. 708–716. Association for Computational Linguistics, Seattle (2007). http://www.aclweb.org/anthology/K/K07/
16. Dalton, J., Dietz, L., Allan, J.: Entity query feature expansion using knowledge base links. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14, pp. 365–374. Association for Computing Machinery, New York (2014). http://doi.acm.org/10.1145/2600428.2609628
17. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In: Proceedings of the 12th International Conference on World Wide Web, WWW '03, pp. 178–186. Association for Computing Machinery, New York (2003). http://doi.acm.org/10.1145/775152.775178
18. Dunn, H.L.: Record linkage. Am. J. Public Health and the Nations Health **36**(12), 1412–1416 (1946). https://doi.org/10.2105/AJPH.36.12.1412. PMID: 18016455
19. Elango, P.: Coreference resolution: A survey. Technical Report, University of Wisconsin, Madison, WI (2005)
20. Ferragina, P., Scaiella, U.: Fast and accurate annotation of short texts with Wikipedia pages. IEEE Softw. **29**(1), 70–75 (2012). http://dx.doi.org/10.1109/MS.2011.122
21. Gottipati, S., Jiang, J.: Linking entities to a knowledge base with query expansion. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pp. 804–813. Association for Computational Linguistics, Stroudsburg (2011). http://dl.acm.org/citation.cfm?id=2145432.2145523
22. Guha, R., McCool, R.: Tap: a semantic web test-bed. Web Semant. Sci. Serv. Agents on the World Wide Web **1**(1), 81–87 (2003). https://doi.org/10.1016/j.websem.2003.07.004. http://www.sciencedirect.com/science/article/pii/S1570826803000064
23. Guo, S., Chang, M.W., Kiciman, E.: To link or not to link? a study on end-to-end tweet entity linking. In: Vanderwende, L., III, H.D., Kirchhoff, K. (eds.) Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, 9–14 June 2013, Westin Peachtree Plaza Hotel, Atlanta, pp. 1020–1030. The Association for Computational Linguistics (2013). http://aclweb.org/anthology/N/N13/N13-1122.pdf
24. Hasibi, F., Balog, K., Bratsberg, S.E.: Entity linking in queries: tasks and evaluation. In: Proceedings of the 2015 International Conference on The Theory of Information Retrieval, ICTIR '15, pp. 171–180. Association for Computing Machinery, New York (2015). http://doi.acm.org/10.1145/2808194.2809473
25. Hasibi, F., Balog, K., Bratsberg, S.E.: Entity linking in queries: efficiency vs. effectiveness. In: Jose, J.M., Hauff, C., Altingövde, I.S., Song, D., Albakour, D., Watt, S.N.K., Tait, J. (eds.) Proceedings of the 39th European Conference on IR Research Advances in Information Retrieval, ECIR 2017, Aberdeen, 8–13 April 2017. Lecture Notes in Computer Science, vol. 10193, pp. 40–53 (2017)

26. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: EMNLP, pp. 782–792. Association for Computational Linguistics, Seattle (2011). http://www.aclweb.org/anthology/D11-1072
27. Hoffart, J., Seufert, S., Nguyen, D.B., Theobald, M., Weikum, G.: Kore: keyphrase overlap relatedness for entity disambiguation. In: Chen, X.-W., Lebanon, G., Wang, H., Zaki, M.J. (eds.) 21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, 29 October–02 November 2012, pp. 545–554. Association for Computing Machinery, New York (2012). http://dl.acm.org/citation.cfm?id=2396761
28. Huang, J., Treeratpituk, P., Taylor, S.M., Giles, C.L.: Enhancing cross document coreference of web documents with context similarity and very large scale text categorization. In: Huang, C.R., Jurafsky, D. (eds.) COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23–27 August 2010, Beijing, pp. 483–491. Tsinghua University Press (2010). http://aclweb.org/anthology/C/C10/
29. Khalid, M.A., Jijkoun, V., de Rijke, M.: The impact of named entity normalization on information retrieval for question answering. Springer, New York (2009). http://dare.uva.nl/record/297954
30. Kulkarni, S., 0003, A.S., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of Wikipedia entities in web text. In: IV, J.F.E., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, 28 June–1 July, 2009, pp. 457–466. Association for Computing Machinery, New York (2009). http://doi.acm.org/10.1145/1557019.1557073
31. Lin, T., Pantel, P., Gamon, M., Kannan, A., Fuxman, A.: Active objects: Actions for entity-centric search. In: World Wide Web. Association for Computing Machinery, New York (2012). http://research.microsoft.com/apps/pubs/default.aspx?id=161389
32. Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., Lu, Y.: Entity linking for tweets. In: ACL (1), pp. 1304–1311. The Association for Computer Linguistics (2013). http://aclweb.org/anthology/P/P13/
33. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, pp. 1–8. Association for Computing Machinery, New York (2011)
34. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Silva, M.J., Laender, A.H.F., Baeza-Yates, R.A., McGuinness, D.L., Olstad, B., Olsen, Ø.H., Falcão, A.O. (eds.) Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, 6–10 November 2007, pp. 233–242. Association for Computing Machinery, New York (2007). http://doi.acm.org/10.1145/1321440.1321475
35. Mohit, B.: Named entity recognition, pp. 221–245 (2014). https://doi.org/10.1007/978-3-642-45358-8_7
36. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Trans. Assoc. Comput. Linguist. **2**, 231–244 (2014). https://transacl.org/ojs/index.php/tacl/article/view/291
37. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticæ Investigationes **30**(1), 3–26 (2007). http://www.jbe-platform.com/content/journals/10.1075/li.30.1.03nad
38. Nagarajan, M., Wilkins, A.D., Bachman, B.J., Novikov, I.B., Bao, S., Haas, P.J., Terrón-Díaz, M.E., Bhatia, S., Adikesavan, A.K., Labrie, J.J., Regenbogen, S., Buchovecky, C.M., Pickering, C.R., Kato, L., Lisewski, A.M., Lelescu, A., Zhang, H., Boyer, S., Weber, G., Chen, Y., Donehower, L.A., Spangler, W.S., Lichtarge, O.: Predicting future scientific discoveries based on a networked analysis of the past literature. In: Cao, L., Zhang, C., Joachims, T., Webb, G.I., Margineantu, D.D., Williams, G. (eds.) Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, 10–13 August 2015, pp. 2019–2028. Association for Computing Machinery, New York (2015). http://dl.acm.org/citation.cfm?id=2783258

39. Newcombe, H.B., Kennedy, J.M., Axford, S.J., James, A.P.: Automatic linkage of vital records. Science **130**(3381), 954–959 (1959). http://science.sciencemag.org/content/130/3381/954
40. Pang, B., Kumar, R.: Search in the lost sense of "query": question formulation in web search queries and its temporal changes. In: ACL (Short Papers), pp. 135–140. The Association for Computer Linguistics (2011). http://www.aclweb.org/anthology/P11-2024
41. Popescu, O.: Dynamic parameters for cross document coreference. In: Huang, C.R., Jurafsky, D. (eds.) COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23–27 August 2010, Beijing, pp. 988–996. Chinese Information Processing Society of China (2010). http://aclweb.org/anthology/C/C10/C10-2114.pdf
42. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: Proceedings of the 19th International Conference on World Wide Web, WWW '10, pp. 771–780. Association for Computing Machinery, New York (2010). http://doi.acm.org/10.1145/1772690.1772769
43. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to Wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT '11, vol. 1, pp. 1375–1384. Association for Computational Linguistics, Stroudsburg (2011). http://dl.acm.org/citation.cfm?id=2002472.2002642
44. Shen, W., Wang, J., Luo, P., Wang, M.: Linking named entities in tweets with knowledge base via user interest modeling. In: Dhillon, I.S., Koren, Y., Ghani, R., Senator, T.E., Bradley, P., Parekh, R., He, J., Grossman, R.L., Uthurusamy, R. (eds.) The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, 11–14 August 2013, pp. 68–76. Association for Computing Machinery, New York (2013). http://dl.acm.org/citation.cfm?id=2487575
45. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW '07, pp. 697–706. Association for Computing Machinery, New York (2007). http://doi.acm.org/10.1145/1242572.1242667
46. Varma, V., Bysani, P., Reddy, K., Reddy, V.B., Kovelamudi, S., Vaddepally, S.R., Nanduri, R., Kumar, N.K., Gsk, S., Pingali, P.: IIIT Hyderabad in guided summarization and knowledge base population. In: TAC. NIST (2010). http://www.nist.gov/tac/publications/2010/papers.html
47. Welty, C., Murdock, J.W., Kalyanpur, A., Fan, J.: A comparison of hard filters and soft evidence for answer typing in Watson. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) The Semantic Web – ISWC 2012, pp. 243–256. Springer, Berlin (2012)
48. West, R., Gabrilovich, E., Murphy, K., Sun, S., Gupta, R., Lin, D.: Knowledge base completion via search-based question answering. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 515–526. Association for Computing Machinery, New York (2014)
49. Zhang, W., Su, J., Tan, C.L., Wang, W.: Entity linking leveraging automatically generated annotation. In: Huang, C.R., Jurafsky, D. (eds.) COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23–27 August 2010, Beijing, pp. 1290–1298. Tsinghua University Press (2010). http://aclweb.org/anthology/C/C10/
50. Zheng, Z., Li, F., Huang, M., Zhu, X.: Learning to link entities with knowledge base. In: HLT-NAACL, pp. 483–491. The Association for Computational Linguistics (2010). http://www.aclweb.org/anthology/N10-1072