# Context Sensitive Entity Linking of Search Queries in Enterprise Knowledge Graphs

Sumit Bhatia and Anshu Jain

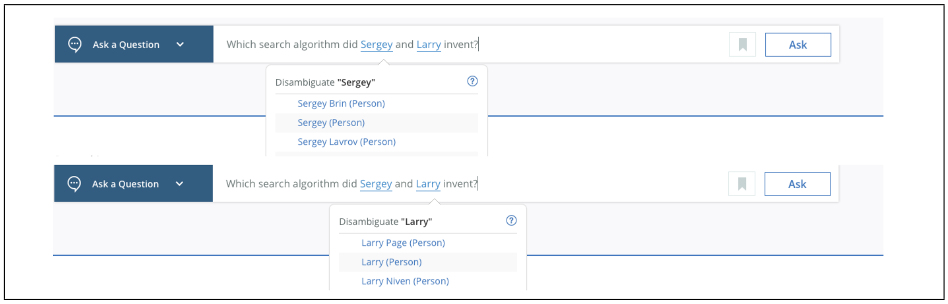
IBM Watson, Almaden Research Centre, San Jose, CA, USA
{sumit.bhatia,anshu.n.jain}@us.ibm.com

**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 work 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 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.

## 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. Further, a large fraction of queries contain a named entity and searchers tend to use more question-queries for complex information needs [2]. 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. Entity linking in search queries utilizes information derived from query logs and open knowledge bases such as DBPedia and Freebase. Such techniques, however, are not suited for enterprise and domain specific search systems such as legal, medical, healthcare, etc. due to very small user bases resulting in small query logs and 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 [3] and systems that use such domain specific knowledge bases [8]. We describe the method used for entity disambiguation and linking as implemented in one such system, *Watson Discovery Advisor*. It 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. 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. [3]. 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, is fast and scalable with an average response time of $\approx \boldsymbol{100ms.}$. Fig 1 shows the working of proposed algorithm in action where top ranked suggestions for named mentions `Sergey` and `Larry` are showed. As will be described in detail in next Section, 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.



Which search algorithm did Sergey and Larry invent?

**Fig. 1:** Entity Suggestions produced by proposed approach using text and entity context in search query.

## 2 Proposed Approach

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. Let $E_T = \{e_{T1}, e_{T2}, \ldots, e_{Tm}\}$ be the set of target entities for $T$. A ranked list of target entities can be constructed 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.

$$P(e_{Ti}|C) = \frac{P(e_{Ti})P(C|e_{Ti})}{P(C)} \propto P(C|e_{Ti}) \tag{1}$$

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. Likewise, assuming all the entities to be equally probable in absence of any context, $P(e_{T_i})$ can be ignored for ranking purposes. Assuming conditional independence for context terms as well as entities in context, we have:

$$P(e_{Ti}|C) \propto P(W_c|e_{Ti}) \times P(E_c|e_{Ti}) = \underbrace{\prod_{w_c \in W_c} P(w_c|e_{Ti})}_{\text{text context}} \times \underbrace{\prod_{e_c \in E_c} P(e_c|e_{Ti})}_{\text{entity context}} \tag{2}$$

**Computing Entity Context Contribution:** The *entity context* factor in equation 2 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:

$$P(e_c|e_{Ti}) = \frac{relCount(e_c, e_{Ti}) + 1}{relCount(e_c) + |E|} \quad (3)$$

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 $e_{Ti}$.

**Computing Text Context Contribution** The *text context* factor in equation 2 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 $e_{Ti}$. This probability can be estimated by using the *mention language model* of $e_{Ti}$ as follows.

$$P(w_c|E_{Ti}) = P(w_c|M_{Ti}) = \frac{\text{no. of times } w_c \text{ appears in mentions of } E_{Ti} + 1}{|M_{Ti}| + N} \quad (4)$$

Here, $N$ is the size of the vocabulary. Since entities are discovered automatically from text, these mentions provide important context information as illustrated in Section 1.

## 3 Evaluation

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 Statistical Information and Relation Extraction (SIRE) toolkit[1]. 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 bases. Our graph contains more than 30 millions entities and 192 million distinct relationships in comparison to 4.5 million entities and 70 million relationships in DBpedia. For evaluating the proposed approach, we use the KORE50 [5] dataset that contains 50 short sentences with highly ambiguous entity mentions. This widely used dataset is considered amongst the hardest dataset for entity disambiguation. 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 [9]. However, it varies for different knowledge bases. Our automatically constructed knowledge base has *2,261 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. The results of our proposed approach and various other state-of-the-art methods for entity

[1] http://ibmlaser.mybluemix.net/siredemo.html

linking on the same dataset are tabulated in Table 1. We note that the performance of our proposed approach is comparable or better than the other approaches, despite dealing with much noisier data. Further, average response time for proposed approach is about 100ms, as we utilize the signals from mention text and relationship information about entities instead of performing complex and time consuming graph operations as in other methods, while not sacrificing on the accuracy.

| **Method** | **Precision** | Method | **Precision** |
|---|---|---|---|
| Joint-DiSER-TopN [1] | 0.72 | DBpedia Spotlight [7] | 0.35 |
| AIDA-2012 [6] | 0.57 | **Proposed Method Accuracy @ Rank 1** | 0.52 |
| AIDA-2013 [5] | 0.64 | **Proposed Method Accuracy @ Rank 5** | 0.65 |
| Wikifier [4] | 0.41 | **Proposed Method Accuracy @ Rank 10** | 0.74 |

**Table 1.** Entity Disambiguation accuracy

## 4 Conclusions

In this paper, we addressed the problem of mapping entity mentions in natural language search queries to corresponding entities in an automatically constructed knowledge graph. We proposed an approach that utilizes the dense graph structure as well as additional context provided by the mention text. Comparative evaluation on a standard dataset with state-of-the-art approaches shows the strengths of our proposed approach in achieving high accuracy with super fast response times. The proposed approach is currently deployed in an enterprise semantic search system called Watson Discovery Advisor and our future work will focus on developing the approach further to utilize user click-feedback for improving the quality of entity suggestions.

## References

1. N. Aggarwal and P. Buitelaar. Wikipedia-based distributional semantics for entity relatedness. In *2014 AAAI Fall Symposium Series*, 2014.
2. A. Aula, R. M. Khan, and Z. Guan. How does search behavior change as search becomes more difficult? In *CHI '10*, pages 35–44, 2010.
3. V. Castelli, H. Raghavan, R. Florian, D.-J. Han, X. Luo, and S. Roukos. Distilling and exploring nuggets from a corpus. In *SIGIR*, pages 1006–1006, 2012.
4. X. Cheng and D. Roth. Relational inference for wikification. In *EMNLP*, pages 1787–1796, 2013.
5. J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *CIKM*, pages 545–554, 2012.
6. J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *EMNLP*, pages 782–792, 2011.
7. P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.
8. M. Nagarajan et al. Predicting future scientific discoveries based on a networked analysis of the past literature. In *KDD '15*, pages 2019–2028, 2015.
9. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW*, pages 697–706, 2007.