Know Thy Neighbors, and More! Studying the Role of Context in Entity Recommendation

Sumit Bhatia
IBM Research AI
New Delhi, India
sumitbhatia@in.ibm.com

Harit Vishwakarma
IBM Research AI
Bangalore, India
harivish@in.ibm.com

ABSTRACT

Knowledge Graphs capture the semantic relations between real-world entities and can thus, allow end-users to explore different *aspects* of an entity of interest by traversing through the edges in the graph. Most of the state-of-the-art methods in entity recommendation are limited in the sense that they allow users to search only in the immediate neighborhood of the entity of interest. This is majorly due to efficiency reasons as the search space increases exponentially as we move further away from the entity of interest in the graph. Often, users perform the search task in the context of an information need and we investigate the role this *context* can play in overcoming the scalability issue and improving knowledge graph exploration. Intuitively, only a small subset of entities in the graph are relevant to a users' interest. We show how can we efficiently select this sub-set by utilizing contextual clues and using graph-theoretic measures to further re-rank this set to offer highly relevant graph exploration capabilities to end-users.

CCS CONCEPTS

• **Information systems → Probabilistic retrieval models**; **Retrieval tasks and goals**; *Web searching and information discovery*; *Content ranking*; *Personalization*; Enterprise information systems;

KEYWORDS

entity Search; entity recommendation; entity retrieval; contextual entity recommendation; contextual exploration; knowledge graph exploration; information discovery

ACM Reference Format:
Sumit Bhatia and Harit Vishwakarma. 2018. Know Thy Neighbors, and More! Studying the Role of Context in Entity Recommendation. In *HT '18: 29th ACM Conference on Hypertext and Social Media, July 9–12, 2018, Baltimore, MD, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3209542.3209548

1 INTRODUCTION

A large fraction of web search queries are entity-centric and contain at least one named entity mention such as names of places, persons, movies, etc. (estimates vary from 40% [27] to 60% [36]). Further, users are often interested in knowing and exploring about a topic of interest rather than obtaining instant answers [12]. To achieve this goal, they perform more exploratory and investigative searches [29] that are often open-ended and gain a better understanding of the topic while interacting with the system [48]. In addition to web search, such entity-oriented exploration tasks are also common in enterprise and domain specific settings [10] such as finding entities related to an entity of interest [9], exploring relations between drugs and genes [19], or studying connections between different criminals and terrorists [42]. In such exploratory tasks, *context* plays a key role in determining the information to be presented to the user. For example, a user interested in knowing more about *Elon Musk* in context of *Tesla Motors* will be interested in a different set of entities than a user who is more interested in *SpaceX*, the space exploration company.

Most of the existing work on such entity-oriented search and exploration (covered in detail in Section 2) have studied entity search or recommendation in context of Web Search [11, 12] where the features derived from query logs and session statistics are used to recommend entities related to the input entity specified by the user; or in *ad-hoc* entity retrieval setting where the focus is on retrieving entities embedded in documents [7, 13, 17, 18]. Such methods rely solely on the textual information present in documents containing entity mentions where the information present is often ambiguous and unstructured and thus, it is harder to utilize the interactions between related entities present in different documents [30].

Recent advancements in semantic search technology have made structured knowledge bases such as DBPedia [5], Yago [44], etc. a critical component of modern information management systems. Many large scale knowledge graphs are often constructed automatically using machine-learned information extraction techniques [4, 15] and can thus also be used in domain specific applications such as finance [40], healthcare [34], cybersecurity [21]. In such knowledge graphs, nodes represent real world concepts or *entities* and their relationships with other such entities are represented as edges in the graph. This structured representation about real world concepts (entities) can help overcome the shortcomings of text-based methods for entity-oriented tasks. For example, in context of recommender systems, variants of personalized page ranks over user and item graphs have been shown to capture indirect relationships in the graph and thus, improving recommendation accuracy [24, 50]. However, one major shortcoming of such graph-based methods is scalability [30] as the number of entities to evaluate increases exponentially with the distance from the seed entity. Consequently, for typical knowledge graphs that contain millions of entities, most graph based methods only work in the immediate neighborhood of

HT '18, July 9–12, 2018, Baltimore, MD, USA

ACM ISBN 978-1-4503-5427-1/18/07…$15.00
https://doi.org/10.1145/3209542.3209548

the input entity or at most up to one hop neighborhood [2, 3, 6, 25] ignoring the useful information present in the rest of the graph.

We posit that the *context* in which the user performs a search or exploration task can be utilized to overcome the shortcomings of existing graph based methods in terms of scalability. Specifically, we argue that the contextual information can be employed to increase the search space over the whole graph, instead of just the direct neighborhood of the input entity. If we can filter the entities of the graph by their relevance to the context, graph structure based approaches can then be efficiently used to re-rank this much smaller, yet highly relevant sub-set of the graph. To test this hypothesis, we focus on the problem of entity-oriented search over knowledge graphs where the user is interested in finding entities relevant to an input entity in context of an information need. For example, a user researching about the Turkish Warrior *Timur* might be interested in knowing entities (places, people, etc.) relevant to *Timur's conquest of Persia.* Thus, the user specifies *Timur* as the input entity, and *Timur's conquest of Persia* as the context and the system returns a list of relevant entities to the user.

We describe a probabilistic formulation that takes into account the contextual information of the entities and then combines it with graph structural features to produce a final list of entities ranked by their relevance to the input entity (Section 3). Experiments conducted using a knowledge graph created out of Wikipedia articles and queries selected from the Wikistream dataset (Section 4) showed that *incorporating contextual information does help* – we are able to find lot more relevant entities beyond the direct neighborhood of the input entity. Further, a combination of graph-based features and contextual information also helps in pushing more relevant entities to the top of result list.

2 RELATED WORK

2.1 Entity Relatedness and Finding Related Entities

Wikipedia, with its rich semantic data and extensive hyperlink structure, has been extensively used for measuring relatedness between two entities (or concepts). Just as the Google Similarity Distance [16] is defined over Google's web graph, Milne and Witten[31] described a measure for computing entity relatedness by utilizing the hyperlink structure in Wikipedia articles [31] and used it for predicting missing links in Wikipedia [32]. Strube and Ponzetto [43] utilized the category hierarchy as provided by Wikipedia to compute relatedness between two entities.

Text Retrieval Evaluation Conference (TREC) introduced a related entity finding (REF) track [7] with an objective to develop benchmark collections and evaluation measures for *entity-oriented* search tasks. Given an input query and a description of users' search intents, the systems were required to produce a ranked list of homepages representing target entities. The REF track, thus, did not take into account the structured relationships between entities and focused only on the content present in entity homepages. In context of TREC REF task, Bron et al. [13, 14] describe the use of co-occurrence statistics for ranking related entities. Fang and Luo [18] describe a probabilistic model for ranking related entities that utilizes Wordnet concepts for estimating the type information of target entities. Raghavan et al. [37] use the context around entity mentions to build *entity langugae models* and use these models to perform related entity finding task. These methods, however, rely mostly on the textual information present in entity homepages and thus, do not utilize the semantic information otherwise present in a knowledge graph.

2.2 Entity Recommendation in Web Search and Other Information Retrieval Systems:

Blanco et al. [12] study the problem of entity recommendations in Web search. Given an input entity, they used a learning to rank approach to rank entities using co-occurrence based features derived from search query logs, tags in flickr and twitter, in addition to graph theoretic features derived from the graph created out of hyperlinks in web pages. Bin et al. [11] incorporate the click data for entity panes shown to users in their entity recommendation system for web search users. Reinanda et al. [39] identify and extract different aspects of an entity from query logs and use these aspects to improve query recommendations to search users.

As examples of domain-specific applications of utilizing knowledge graphs, Fokoue et al. [19, 20] proposed a framework to predict drug-drug interactions through similarity based link prediction.

In context of semantic knowledge bases, Wang et al. [47] and Zhang et al. [51] proposed time aware entity recommendation methods that are developed on the intuition that relationship between entities evolve over time (e.g. *married* relationship between two persons is valid only for a specific period in time). Tran et al. [45] improve upon such models by recommending topic and time sensitive results. However, like other link prediction and recommendation methods, they also limited their models to direct neighbors of the input entity whereas the focus of present work is to study how context can be utilized to efficiently increase the search space to include entities that may not be directly connected.

3 PROPOSED APPROACH

We first describe the problem setting and present a mathematical formulation of the contextual entity recommendation problem. We then describe our proposed probabilistic framework and describe different components of the framework in detail.

3.1 Problem Formulation

Let $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$ be a knowledge graph with $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ as the set of entities (nodes) and $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ as the set of relationships (edge set). Let $\mathcal{D}$ be the underlying document corpus. For each edge $r \in \mathcal{R}$, $\mathcal{P}_r = \{p_{r1}, p_{r2}, \ldots, p_{rk}\}$, is the set of passages in $\mathcal{D}$ that contain mentions of relationship r. This passage set is generally available for automatically constructed knowledge bases [10, 15] as these methods output the portions of text from which a specific relationship is identified. Even in manually curated knowledge graphs, these passages can be identified by using entity-linking techniques [33].

Next, consider a user who wants to explore this graph. The user specifies a starting query entity e_q and the text context C, and would like to see entities from the graph relevant to entity e_q in context, C. For example, the user may be interested in knowing entities relevant to `Steve Jobs` in context of `pixar animation`. In such a case, an entity like `Steve Wozniak` is not relevant for the user even though it is a very important entity for `Steve Jobs`, whereas, `Edwin Catmull`, a Pixar executive is highly relevant, even though `Steve Jobs` and `Edwin Catmull` are weakly connected in the graph.

Mathematically, having observed the input entity e_q and context C, we are interested in computing the probability of observing a target entity e, i.e, $P(e|e_q, C)$.

Application of Bayes' Theorem yields

$$P(e|e_q, C) \propto P(e)P(e_q, C|e) \tag{1}$$

Here, the denominator $P(e_q, C)$ can be ignored as it is constant for all target entities and will not alter the relative ranking of target entities. Assuming e_q and C to be independent, the above equation can be written as follows.

$$P(e|e_q, C) \propto \underbrace{P(e)}_{\text{entity prior}} \times \underbrace{P(e_q|e)}_{\text{entity affinity}} \times \underbrace{P(C|e)}_{\text{context relevance}} \tag{2}$$

Note that the above formulation clearly separates the probability computation into three components – prior probability of target entity, affinity between target and query entity and relevance of target entity in context C. While the prior and entity affinity components can be computed using structural properties of the graph, context relevance can be computed using the underlying document corpus $\mathcal{D}$. We discuss in detail the choices for the different components of the model in the following sub-sections.

3.2 Entity Prior

This component measures the prior probability of observing the entity e and is independent of the input query e_q and context C. Intuitively, in absence of any input information, an entity that has connections with many other entities in the graph has a higher probability of being observed than an isolated entity. Therefore, we define the entity-prior in terms of degree of each entity as follows:

$$P(e) = \frac{d(e)}{2|\mathcal{R}|} \tag{3}$$

Here $d(e)$ denotes the degree (in-degree + out-degree) of e and $\mathcal{R}$ is the number of edges in the knowledge graph. Note that this is a valid probability distribution and sums to 1 when summed over all the entities, since $\Sigma_e d(e) = 2|\mathcal{R}|$. This prior assigns high score to entities that are connected to a large number of other entities and also helps in reducing the scores of noisy or erroneous entities (present in the graph as a result of imperfect automatic knowledge extraction methods) that typically have very few connections.

3.3 Entity Affinity

Entity affinity captures the likelihood of association between two given entities and is a measure of semantic relatedness between them. We can exploit the knowledge graph to compute the affinities and capture the rich structural information available in the graph.

We assume $P(e_q|e)$ is same as $P(e|e_q)$. This allows us to model `entity-affinity` as a measure of similarity. While the relatedness between two entities in the graph can be captured using multiple ways, we study following three widely studied relatedness measures.

Adamic-Adar (AA): It was originally proposed to predict whether one person is likely to be associated with another in an academic social network constructed from web-pages [2]. Empirically, this measure has been shown to perform better than many other common neighbors based similarity metrics such as common neighbors count, Jaccard and cosine similarities for link-prediction in social networks [26]. It is based on the intuition that if two persons are similar then they will share many common "friends" between them. Moreover, a person who is connected to a few is weighted more than the person connected to many, since connections with such less popular nodes are more informative and discriminative. In terms of a graph, two nodes are highly similar if they have many common neighbors which are not connected to a large number of other nodes. For example, the fact that both `Steve Jobs` and `Bill gates` share `United States` as a common neighbor is not very informative and should not contribute heavily in determining their relatedness as there are many other entities in the graph that have connections with `United States`. Hence, it is important to assign low weights to popular nodes, which is a major shortcoming of some of the other neighborhood based measures like cosine similarity and Jaccard similarity. Formally `Adamic-Adar` similarity $AA(u, v)$ between two nodes $u, v \in \mathcal{E}$ is defined as following:

$$AA(u, v) = \sum_{x \in N(u) \cap N(v)} \frac{1}{\log(|N(x)|)} \tag{4}$$

Here $N(w)$ denotes the set of nodes to which there are outgoing edges from w in the knowledge graph $\mathcal{G}$. Note that it is computationally inexpensive hence can be used with very large graphs easily.

Milne-Witten (MW): It was introduced to compute semantic relatedness between Wikipedia articles using only the hyper-link structure (graph) between the articles [32]. The basic intuition behind this measure is that two Wikipedia articles are topically related if there are many Wikipedia articles that link to both of them. It has been successfully applied to a variety of tasks related to Wikipedia data such as for measuring semantic associativity between Wikipedia concepts for entity disambiguation [38] and entity linking tasks [41].

Formally `Milne-Witten` similarity $MW(u, v)$ between two nodes $u, v \in \mathcal{E}$ is defined as following:

$$MW(u, v) = \frac{\log\Big(\max(|N'(u)|, |N'(v)|)\Big) - \log\Big(|N'(u) \cap N'(v)|\Big)}{\log\Big(|\mathcal{E}|\Big) - \log\Big(\min(|N'(u)|, |N'(v)|)\Big)} \tag{5}$$

Here $N'(w)$ denotes the set of nodes having outgoing edges to w.

Note that just like `Adamic-Adar` and other neighborhood based methods, this measure also relies only on the neighborhood information of input entities and is therefore, easy to compute even on large graphs. However, this limitation prevents use of these measures for computing relatedness of entities that share no common neighbors. Consequently, the search space for these methods is limited to the second-hop neighbors of the query entity, however in practice there could be many relevant entities that lie beyond the second hop neighborhood. Further, since these methods utilize information only about the common neighbors and ignore longer path between entities, these methods are limited in capturing complete structural similarity induced by the graph.

SimRank: To overcome the limitations discussed above, we used the `SimRank` algorithm proposed by Jeh and Widom [22] that is based on the intuition that "two nodes are related if they are related to similar nodes". Note that this recursive definition of SimRank allows us to capture arbitrarily long paths and compute structural similarities between entities that are farther away in the graph. In contrast to the neighborhood based similarity scores it can measure similarities among nodes that don't share any common neighbors. Thus, it allows us to look beyond first and second hop neighbors of a given node and has been empirically applied to a variety of tasks such as predicting links in social networks [26].

Formally, `SimRank` $SR(u, v)$ for any two vertices $u, v \in \mathcal{E}$ is defined as following.

$$SR_{\gamma}(u,v) = \begin{cases} 1, & \text{if } u = v \\ \gamma \cdot \frac{\sum_{x \in N'(u)} \sum_{y \in N'(v)} SR_{\gamma}(x,y)}{|N'(u)|.|N'(v)|}, & \text{otherwise} \end{cases} \tag{6}$$

Here, $\gamma \in (0, 1)$ is a constant called decay factor, which assigns a lower weight to far-away nodes in the graph. Typically $\gamma = 0.8$ has been used in literature. The above equation defines SimRank in a recursive fashion such that the SimRank between two nodes u and v is computed as a function of pair-wise SimRank score computed between their neighbors. The base case, as represented by the first part of above equation, denotes that an entity is maximally similar to itself.

SimRank Computation: As is evident from Equation 6, recursive computation of SimRank is computationally inefficient and it is infeasible to compute it for large graphs. As a result, fast and scalable approximation algorithms have been proposed in literature for efficient SimRank computation. We used one such recently proposed single pair SimRank algorithm [23] based on random walks and monte carlo simulations for our implementation. Its time complexity is $O(TR)$, where R is the number of random walks simulated and T is the maximum steps up to which the walks are performed.

Normalization to Probabilities: The above graph based affinity scores are not valid probabilities hence we have to normalize them appropriately. Let $S(x, y)$ be a similarity measure between nodes x and y. Then a normalized version is as follows:

$$P(x|y) = \frac{1 + S(x,y)}{\sum_{z \in \mathcal{E}} S(z,y) + |\mathcal{E}|} \tag{7}$$

Note here that the denominator requires us to compute the similarity function over all pairs of entities x and y which may not always be feasible due to the size of the graph. However, we also note that the denominator remains same for all the entities in the graph, and here can be ignored as we are only interested in relative ranking of the entities.

3.4 Context Relevance

This component measures the relevance of the target entity to the context in which the search/exploration task is being performed. As discussed previously, this is a crucial component of our proposed framework as it can help us in identifying a shortlist of contextually relevant entities that can then be re-ranked to produce the final result list. In our problem setting, the context is represented as a set of terms input by the user (Section 3). Assuming that the context terms are observed independently of each other, the component $P(C|e)$ can be estimated as follows:

$$P(C|e) = \prod_{c \in C} P(c|e) \tag{8}$$

Here, $c \in C$ are the constituent terms of context C.

In order to compute the probability of the term c given the entity e, we built a *context document* for each entity in the graph by utilizing the relationship passage sets described in Section 3. Intuitively, the probability of observing a context term given an entity is higher if that term appears frequently with mentions of the input entity in the underlying corpus. If $\mathcal{R}_e = \{r_{e1}, r_{e2}, \ldots, r_{en}\}$ be the set of relationships in which entity e is involved, and $P_{r_{ei}}$ is the set of passages from which r_{ei} was extracted, the context document for entity e is defined as follows:

$$CD(e) = \bigcup_{r_{ei} \in \mathcal{R}_e} P_{r_{ei}} \tag{9}$$

Thus, a context document for an entity is the concatenation of all the passages from the text corpus from which a relation involving the entity was extracted. Once the context document of the entity is build, the probability of observing a term given the entity can be estimated using a unigram language model for the context document [28, Chapter 12] as follows:

$$P(c|e) = P(c|CD(e)) = \frac{tf(c) + 1}{|CD(e)| + |V|} \tag{10}$$

Here, $tf c$ is the term frequency of term c in the context document and $|V|$ is the total number of terms in the vocabulary. Note that the factor of one in numerator is added for smoothing purposes to prevent zero probabilities for terms not present in the context document [28, Chapter 12].

As an example, consider the entity `Steve jobs` that has many relationships with different Apple products (`iPhone, iPod, iPad, Macintosh`, etc.). Words occurring in passages from which these relationships are extracted are representative of different *contexts* in which `Steve jobs` appears (such as design, development, invention, functioning, etc.). Likewise, `Steve Jobs` has many relationships with different executives like `Tim Cook, Eddie Cue, Jonathan Ive`, etc. and words occurring in relationship passages with these entities will be different than those occurring with Apple products. Thus, the context document for `Steve Jobs` will capture different terms representative of different *contexts* relevant to `Steve Jobs`. Also note that these words do not correspond to named entities and hence, are not present in the graph but will be captured in the context document for `Steve Jobs`.

Method	Score Function
Context (C)	$\log P(C\|e)$
Degree (D)	$\log P(e)$
D+Adamic-Adar(AA)	$\log P(e) + \log(1 + AA(e_q, e))$
D+Milne-Witten(MW)	$\log P(e) + \log(1 + MW(e_q, e))$
D + SimRank (SR)	$\log P(e) + \log(1 + SR(e_q, e))$
D + C + AA	$\log P(e) + \log(1 + AA(e_q, e)) + \log P(C\|e)$
D + C + MW	$\log P(e) + \log(1 + MW(e_q, e)) + \log P(C\|e)$
D + C + SR	$\log P(e) + \log(1 + SR(e_q, e)) + \log P(C\|e)$

Table 1: Summary of methods with their corresponding score functions for ranking.

3.5 Final Scoring Function and Variations:

Finally, we plug-in the above prior, entity-affinity and context relevance scores in Equation 2 and take log to arrive at the following final ranking function:

$$\text{score}(e|e_q, C) = \log P(e) + \log(1 + S(e_q|e)) + \log P(C|e) \qquad (11)$$

This scoring function allows us to obtain different variations which are summarized in Table-1. There are three sections in the table, first is just the context based method and it doesn't use any information from the knowledge graph and relies completely on the results from the text corpus, the second section is of purely graph based methods that use only the graph based scores to rank the entities. The third group combines the above two, i.e. it combines both the graph based features and context relevance. In the next Section, we use these variations to study the impact of different components of the ranking function (Equation 11).

4 EXPERIMENTS

In this section we describe the data and query set we used for evaluation. We provide details about the knowledge graph and entity language model construction. We describe a new dataset for this task that we constructed using the WikiStream dataset [49]. Implementation details of the scores mentioned in the Equation 11 are discussed. We show comparison of methods utilizing only the context information, graph based scores, and combinations of these using well known performance measures. We also study the distribution of relevant entities at different path lengths from the query entity to understand the importance of entities at different distances.

4.1 Data Description

4.1.1 ***Text Corpus:*** We use dump of the English Wikipedia as our background text corpus $\mathcal{D}$ as it is a snapshot of the general open domain knowledge about the World. It has around 5 million articles and is used to construct our knowledge graph and entity context documents as described next.

Knowledge Graph: We use a semantic graph constructed from the text of all articles in Wikipedia by automatically extracting the entities and their relations by using Statistical Information and Relation Extraction (SIRE) toolkit [15]. 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.

Entity Context Documents: We construct the context documents for all 30 million+ extracted entities in our knowledge graph and indexed them using the Indri Language Modeling Toolkit as provided by the Lemur project [1]. **Indexing Entity Context Documents:** To efficiently search the documents relevant in a given context we use Indri search engine of the Lemur project. Indri is efficient, scalable and gives highly accurate search results [46]. In addition Indri also gives the likelihood score for each result it returns, which we can use for `context-relevance` score. In the indexing process we use a standard stopwords list as provided by the Onix text retrieval toolkit[1] and krovetz stemmer as implemented in Indri.

4.2 Query Set

We create the query set by using the recently released WikiStream dataset [49] following an approach similar to the one followed by Tran et al. [45]. Entity mentions on a Wikipedia page are often linked to their respective Wikipedia pages and users often click on these linked Wikipedia pages to read more about these related entities. Further, since the articles in Wikipedia are often categorized into sub-topics, an entity link mentioned in a specific section of the article and clicked frequently by the users can be considered a proxy for relevance of the clicked entity to the source entity in context of the sub-topic/aspect. For example, "Pixar and Disney" section on Steve Jobs' Wikipedia article contains links to Lucasfilm, Bob Iger, Michael Eisner, etc. – entities relevant to Steve Jobs in context of Disney and Pixar, even though, they may not be deemed relevant otherwise. The WikiStream dataset is created by processing Wikipedia request logs for the month of February 2015 and consists of $< referer, resource >$ pairs where $resource$ is a Wikipedia article and a $referer$ could be another Wikipedia article, or some other external source (request coming from search engines, other web pages, etc.). We extracted all the click pairs from the click stream logs where the referrer and resource were Wikipedia entities and mapped the click pairs with the title of the sub-section where the entity was mentioned giving us $< inputentity, context >$ pairs to use as query and an associated list of clicked entities as our answer set. Hence we have a collection of tuples (query entity, context, relevant entities) and we randomly select a subset of 50 such tuples and refer it by `WikiContext` dataset in the following sections. Table 2 presents some example queries from our dataset. The complete list of queries, context, and relevant entities used in our experiments can be accessed at http://sumitbhatia.net/source/datasets.html.

[1] http://www.lextek.com/manuals/onix/stopwords1.html

Input Entity	Context	Example Answer Entities
Lee H. Oswald	John F. Kennedy and J. D. Tippit Shootings	James Tague, John Connally, Texas Theatre
Timur	Campaign Against The Tughlaq Dynasty	Delhi Sultanate, Sultan Nasir-U Din Mehmud
Martina Hingis	Injuries and Hiatus From Tennis	Williams Sisters, Hopman Cup
Art Modell	As Principal Owner of Baltimore Ravens 1996-2004	Ted Marchibroda, Brian Billick
Art Modell	As Cleveland Browns owner 1961–1995	Paul Brown, Blanton Collie

Table 2: Example queries from the WikiContext dataset.

4.3 Scores Computation

4.3.1 ***Context Score:*** We search for relevant documents (entities) in the given context C by using C as input query to Indri. It returns a list of documents most relevant to the given query but note that in the corpus each indexed document corresponds to an entity. As a result, we obtain a ranked list of contextually relevant entities. Indri also provides log probability score for each output entry denoting its relevance to the query. We use this score as the Context relevance score defined in equation 2. From the results returned by Indri we select top-100 entities and compute the entity-affinity scores for these shortlisted entities.

4.3.2 ***Graph Scores:*** We compute graph scores between the query entity (source entity) and the target entities (shortlisted based on their relevance to context). Similarity measures like Adamic-Adar, Milne-Witten etc. can be computed easily by simple neighborhood queries. However SimRank computation requires backward random walks starting from the source and target entities each. Since it is a Monte Carlo based method, we have to take many samples of the walk in order to get a good approximation. We store the full graph in-memory so that random walks could be simulated efficiently. We use $\gamma = 0.8$, number of random walks (samples) $R = 200$ and maximum distance $T = 10$, for SimRank computation between the source and target entities.

4.4 Evaluation Protocol

We first retrieve top-100 entities based on the context from the entity docs indexed using Indri. This gives us candidate entities relevant to the context and limits the search space as well. We compute the degree score for these entities and the graph based similarity scores between the target entity and the candidate entities. The candidate entities are re-ranked using different combinations of scores listed in Table- 1. We then evaluate the quality of results by comparing against the automatically obtained ground truth and relevance scores obtained from human evaluators. We are interested in Top-5, Top-10 and Top-25 final entities obtained after re-ranking the candidate set.

We report Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), Precision and Recall @K to evaluate the effectiveness of context and graph based entity retrieval components. We first re-rank the top-100 candidate entities by different scores and then compute the measures by taking top-k results of the re-ranked list. We report the average of each performance metric over all the queries in the given query set.

4.5 Results and Discussions

We evaluate the quality of results obtained from different methods and also study the relevance of entities beyond neighborhood. We evaluate results on this dataset against the automated ground truth as well as the ground truth obtained from manual labeling for all 50 queries.

4.5.1 ***Automated Ground Truth:***
Note that in the automatically extracted ground truth, graded relevance judgments are not available. Hence we can't compute NDCG in this case and report Precision, Recall and MRR in Table-3. It can be observed from the table that the combination (D+C+SR) outperforms pure Context and pure graph based methods across all metrics. Note that the pure graph based methods lack the context relevance scores hence their performance is not as good as just the context based method as they produce a *static* ranking that remains same for different contexts. The context based method, while capable of finding contextually relevant entities, suffers from not utilizing the similarity information induced by rich graph structure. Augmenting the context based method with graph structural information produces consistently better results providing support for our hypothesis that incorporating contextual clues to graph based similarity measures can help retrieve more relevant entities.

4.5.2 ***Manual Ground Truth:***
In the automatically constructed ground truth created from the WikiStream dataset, relevance judgments for all the entities are not available for all the entities retrieved by different methods. Therefore, we obtained relevance labels for top 100 shortlisted entities for each query from two human judges for better evaluation and comparison of the results obtained by different methods. The judges were presented with the query entity, context and the Top-100 shortlisted entities in random order and were asked to assign labels from $\{0 : \text{irrelevant}, 1 : \text{relevant}, 2 : \text{highly relevant}\}$ for each result entity. In case of disagreements, the final judgments were aggregated by selecting the minimum value of the two judgments for each result entity. Table-4 reports the numbers on these aggregated relevance labels. Once again, similar observations to Table 3 can be made. We note that the method (D+C+SR) consistently outperforms other methods – it not only finds more relevant entities, it is able to produce a better ranking as indicated by higher NDCG values. Moreover, the gap in performance when compared to other methods is also significant.

4.5.3 ***Per Query Performance Comparison***
. Next, we study how the best performing method (D+C+SR) performs against the second best method D+C+AA and just the Context based method C. We take the difference between the performance metrics (Δ Precision @10, ΔRecall @10, ΔNDCG @10) obtained by D+C+SR and C, D+C+AA for all 50 queries in the dataset. We sort

	@5			@10			@25		
	P	R	MRR	P	R	MRR	P	R	MRR
Context (C)	0.116	0.116	0.212	0.104	0.208	0.241	0.062	0.312	0.251
Degree (D)	0.080	0.080	0.170	0.064	0.128	0.195	0.054	0.268	0.212
D + Adamic Adar(AA)	0.096	0.096	0.194	0.082	0.164	0.226	0.063	0.316	0.239
D + Milne Witten(MW)	0.080	0.080	0.169	0.060	0.120	0.189	0.051	0.256	0.208
D + SimRank (SR)	0.112	0.112	0.195	0.094	0.188	0.227	0.063	0.316	0.237
D + C + AA	0.100	0.100	0.215	0.094	0.188	0.253	**0.066**	0.328	0.269
D + C + MW	0.088	0.088	0.204	0.074	0.148	0.227	0.060	0.300	0.247
D + C + SR	**0.124**	**0.124**	**0.225**	**0.112**	**0.224**	**0.262**	**0.066**	**0.332**	**0.270**

Table 3: Wiki Results with automated ground truth

	@5				@10				@25			
	P	R	NDCG	MRR	P	R	NDCG	MRR	P	R	NDCG	MRR
Context (C)	0.212	0.081	0.105	0.277	0.236	0.187	0.170	0.313	0.210	0.410	0.270	0.317
Degree (D)	0.164	0.070	0.083	0.289	0.194	0.166	0.142	0.327	0.189	0.407	0.243	0.339
D + Adamic Adar(AA)	0.208	0.101	0.118	0.355	0.228	0.217	0.189	0.379	0.206	0.439	0.286	0.386
D + Milne Witten(MW)	0.156	0.068	0.079	0.275	0.188	0.163	0.137	0.315	0.182	0.382	0.232	0.325
D + SimRank (SR)	0.264	0.151	0.171	0.384	0.244	0.255	0.233	0.412	0.217	0.455	0.328	0.414
D + C + AA	0.224	0.110	0.137	0.353	0.230	0.219	0.204	0.382	0.213	0.444	0.304	0.387
D + C + MW	0.180	0.074	0.094	0.330	0.198	0.172	0.151	0.362	0.200	0.425	0.259	0.374
D + C + SR	**0.308**	**0.170**	**0.192**	**0.398**	**0.278**	**0.275**	**0.258**	**0.423**	**0.226**	**0.470**	**0.349**	**0.426**

Table 4: Results on WikiContext Dataset with manually obtained ground truth.

these Δ values and plot them on bar plots, shown in Figure-1. These values are computed on the manually obtained ground truth. We observe that `D+C+SR` performs better than the `C` and `D+C+AA` methods in terms of precision and recall @10 for around 20 queries and around 30 queries for NDCG @10. Thus, on an average, more queries are benefited by combining contextual and graph based measures than using either of them in isolation. We also note that in general, gains for queries that benefit from the combination are more than the loss in performance for few queries increasing the overall performance. We also observe that while both *D+C+AA* and *D+C+SR* achieve better performance when compared with just using the context, the gains are more prominent for the *D+C+SR* method. Given that Adamic Adar (AA) impacts only the entities that share common neighbors with the input entity while SimRank has no such limitation, better performance achieved by SimRank once again lends weight to the importance of going beyond the 1-hop or 2-hop neighborhood of input entity for finding contextually relevant entities.

4.5.4 *Contributions of non-neighbors:*

Next, in order to understand how many relevant entities are found beyond the immediate neighborhood, we compute the lengths of shortest paths from the query entity to all the relevant entities in the contextually relevant shortlist (top-100 entities). We then re-rank these entities by the SimRank based *D+C+SR* method and compute the distribution of different path lengths for top 10, 50 and 100 positions. The results are summarized in Figure-2. We observe that there exist a significant number of relevant entities beyond the immediate neighbors (Path Length = 1). Methods like `Adamic-Adar` and `Milne-Witten` are based on common neighbors and hence they can find entities only till path length = 2. As we see in the plot there is significant fraction of relevant entities at path length greater than 2 and this increases as we increase the value of K. For $K = 100$, only 30% of relevant entities are found in the immediate neighborhood and about 20% of relevant entities lie at a path length of 3 – a significant number that is never evaluated by traditional neighborhood methods.

5 CONCLUSIONS AND FUTURE WORK

We studied the problem of finding relevant entities that the end-user might want to explore given an input entity and context specified as text keywords. We argued that utilizing this context information can help overcome the scalability problem of standard graph based approaches of entity recommendation. We showed how context can be employed in producing a focused shortlist of relevant entities by performing a fast search over the complete graph and then computing graph based features only on this much smaller set. Experiments conducted over a knowledge graph created out of Wikipedia articles showed that by utilizing contextual information helps retrieve more relevant entities, and combining with graph features improves the ranking performance. We also

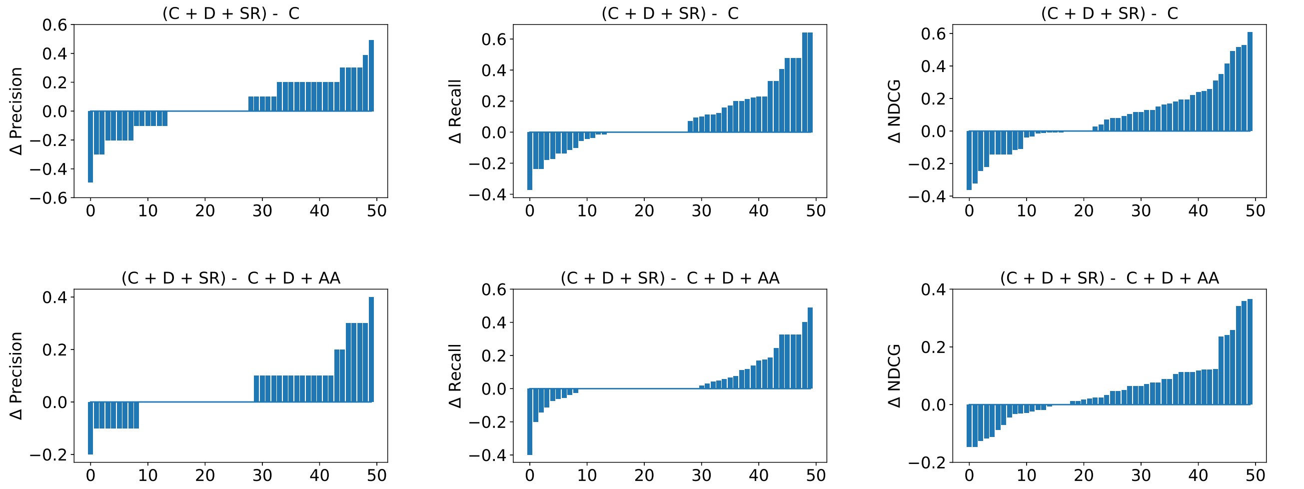

Figure 1: Bar plot showing performance difference per query. X-axis represents queries and bars represent the difference between Precision, Recall, NDCG @10 in the order left to right. First row shows the performance difference between ensemble of Context, Degree Prior and SimRank (C + D + SR) and only the Context (C) while the second row shows comparison against ensemble with Adamic Adar (AA)

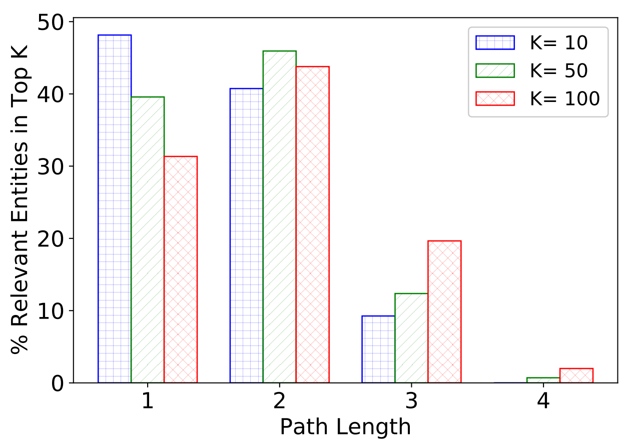

Figure 2: Bar plot showing percentage of relevant entities at different path lengths for Top-K entities ranked by (D + C + SR)

found that a significant fraction of relevant entities lie outside the immediate neighborhood of input entities, thus corroborating our initial hypothesis. Since most of the existing work on entity recommendation has focused on immediate neighborhood of the input entity, explaining how the recommended entities are connected to the input entity was not crucial. Our future work will focus on developing methods for explaining how the entities that are not directly connected to input entities are relevant to the input entity. For this, both graph based methods (such as path ranking [3, 35]) can be utilized as well as textual explanations [8] by utilizing the relationships passages could be produced.

REFERENCES

[1] 2000. The Lemur Toolkit. https://www.lemurproject.org/. Accessed: 2018-01-31.
[2] Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the Web. *Social Networks* 25, 3 (2003), 211 – 230. https://doi.org/10.1016/S0378-8733(03)00009-1
[3] Nitish Aggarwal, Sumit Bhatia, and Vinith Misra. 2016. Connecting the Dots: Explaining Relationships Between Unconnected Entities in a Knowledge Graph. In *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers.* 35–39. https://doi.org/10.1007/978-3-319-47602-5_8
[4] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 344–354.
[5] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web.* Springer, 722–735.
[6] Lars Backstrom and Jure Leskovec. 2011. Supervised Random Walks: Predicting and Recommending Links in Social Networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11).* ACM, New York, NY, USA, 635–644. https://doi.org/10.1145/1935826.1935914
[7] Krisztian Balog, Arjen P de Vries, Pavel Serdyukov, Paul Thomas, and Thijs Westerveld. 2009. Overview of the TREC 2009 entity track. In *In Proceedings of the Eighteenth Text REtrieval Conference.*
[8] Sumit Bhatia, Purusharth Dwivedi, and Avneet Kaur. 2018. Tell Me Why Is It So? Explaining Knowledge Graph Relationships by Finding Descriptive Support Passages. *CoRR* abs/1803.06555 (2018). arXiv:1803.06555 http://arxiv.org/abs/1803.06555
[9] Sumit Bhatia, Alok Goel, Elizabeth Bowen, and Anshu Jain. 2016. Separating Wheat from the Chaff - A Relationship Ranking Algorithm. In *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers (Lecture Notes in Computer Science)*, Harald Sack, Giuseppe Rizzo 0002, Nadine Steinmetz, Dunja Mladenic, Sören Auer, and Christoph Lange 0002 (Eds.), Vol. 9989. 79–83.
[10] Sumit Bhatia, Nidhi Rajshree, Anshu Jain, and Nitish Aggarwal. 2017. Tools and Infrastructure for Supporting Enterprise Knowledge Graphs. In *International Conference on Advanced Data Mining and Applications.* Springer, 846–852.
[11] Bin Bi, Hao Ma, Bo-June Paul Hsu, Wei Chu, Kuansan Wang, and Junghoo Cho. 2015. Learning to recommend related entities to search users. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.* ACM, 139–148.
[12] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. 2013. Entity Recommendations in Web Search. In *Proceedings of the 12th International*

Semantic Web Conference - Part II (ISWC '13). Springer-Verlag New York, Inc., New York, NY, USA, 33–48. https://doi.org/10.1007/978-3-642-41338-4_3

[13] Marc Bron, Krisztian Balog, and Maarten De Rijke. 2010. Ranking related entities: components and analyses. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 1079–1088.

[14] Marc Bron, Krisztian Balog, and Maarten de Rijke. 2009. *Related entity finding based on co-occurrence*. Technical Report. AMSTERDAM UNIV (NETHERLANDS).

[15] Vittorio Castelli, Hema Raghavan, Radu Florian, Ding-Jung Han, Xiaoqiang Luo, and Salim Roukos. 2012. Distilling and Exploring Nuggets from a Corpus. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 1006–1006. https://doi.org/10.1145/2348283.2348431

[16] Rudi L Cilibrasi and Paul MB Vitanyi. 2007. The google similarity distance. *IEEE Transactions on knowledge and data engineering* 19, 3 (2007).

[17] Gianluca Demartini, Tereza Iofciu, and Arjen P. De Vries. 2010. Overview of the INEX 2009 Entity Ranking Track. In *Proceedings of the Focused Retrieval and Evaluation, and 8th International Conference on Initiative for the Evaluation of XML Retrieval (INEX'09)*. Springer-Verlag, Berlin, Heidelberg, 254–264. http://dl.acm.org/citation.cfm?id=1881065.1881096

[18] Yi Fang and Luo Si. 2015. Related entity finding by unified probabilistic models. *World Wide Web* 18, 3 (2015), 521–543.

[19] Achille Fokoue, Oktie Hassanzadeh, Mohammad Sadoghi, and Ping Zhang. 2016. Predicting Drug-Drug Interactions Through Similarity-Based Link Prediction Over Web Data. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 175–178. https://doi.org/10.1145/2872518.2890532

[20] Achille Fokoue, Mohammad Sadoghi, Oktie Hassanzadeh, and Ping Zhang. 2016. Predicting Drug-Drug Interactions Through Large-Scale Similarity-Based Link Prediction. In *Proceedings of the 13th International Conference on The Semantic Web. Latest Advances and New Domains - Volume 9678*. Springer-Verlag New York, Inc., New York, NY, USA, 774–789. https://doi.org/10.1007/978-3-319-34129-3_47

[21] Michael Iannacone, Shawn Bohn, Grant Nakamura, John Gerth, Kelly Huffer, Robert Bridges, Erik Ferragut, and John Goodall. 2015. Developing an Ontology for Cyber Security Knowledge Graphs. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference (CISR '15)*. ACM, New York, NY, USA, Article 12, 4 pages.

[22] Glen Jeh and Jennifer Widom. 2002. SimRank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Press, New York, NY, USA, 538–543. https://doi.org/10.1145/775047.775126

[23] Mitsuru Kusumoto, Takanori Maehara, and Ken-ichi Kawarabayashi. 2014. Scalable Similarity Search for SimRank. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD '14)*. ACM, New York, NY, USA, 325–336. https://doi.org/10.1145/2588555.2610526

[24] Sangkeun Lee, Sang-il Song, Minsuk Kahng, Dongjoo Lee, and Sang-goo Lee. 2011. Random walk based entity ranking on graph for multidimensional recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 93–100.

[25] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. 2008. Microscopic Evolution of Social Networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, New York, NY, USA, 462–470. https://doi.org/10.1145/1401890.1401948

[26] David Liben-Nowell and Jon Kleinberg. 2003. The Link Prediction Problem for Social Networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM '03)*. ACM, New York, NY, USA, 556–559. https://doi.org/10.1145/956863.956972

[27] Thomas Lin, Patrick Pantel, Michael Gamon, Anitha Kannan, and Ariel Fuxman. 2012. Active Objects: Actions for Entity-Centric Search. In *World Wide Web*. ACM. http://research.microsoft.com/apps/pubs/default.aspx?id=161389

[28] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[29] Gary Marchionini. 2006. Exploratory Search: From Finding to Understanding. *Commun. ACM* 49, 4 (April 2006), 41–46.

[30] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. 1400–1409. http://aclweb.org/anthology/D/D16/D16-1147.pdf

[31] David Milne and Ian H. Witten. 2008. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *In Proceedings of AAAI 2008*.

[32] David Milne and Ian H. Witten. 2008. Learning to Link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. ACM, New York, NY, USA, 509–518. https://doi.org/10.1145/1458082.1458150

[33] Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics* 2 (2014), 231–244.

[34] Meenakshi Nagarajan et al. 2015. Predicting Future Scientific Discoveries Based on a Networked Analysis of the Past Literature. In *KDD (KDD '15)*. 2019–2028.

[35] Giuseppe Pirrò. 2015. Explaining and Suggesting Relatedness in Knowledge Graphs. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*. 622–639.

[36] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. 2010. Ad-hoc Object Retrieval in the Web of Data. In *WWW '10*. 771–780.

[37] Hema Raghavan, James Allan, and Andrew McCallum. 2004. An Exploration of Entity Models, Collective Classification and Relation Description. In *Proceedings of the Second International Workshop on Link Analysis and Group*. 1–10.

[38] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1375–1384.

[39] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2015. Mining, ranking and recommending entity aspects. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 263–272.

[40] Tong Ruan, Lijuan Xue, Haofen Wang, Fanghuai Hu, Liang Zhao, and Jun Ding. 2016. Building and Exploring an Enterprise Knowledge Graph for Investment Analysis. In *International Semantic Web Conference*. Springer, 418–436.

[41] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. LINDEN: Linking named entities with knowledge base via semantic knowledge. (04 2012).

[42] Amit Sheth, Boanerges Aleman-Meza, I Budak Arpinar, Clemens Bertram, et al. 2005. Semantic association identification and knowledge discovery for national security applications. *Journal of Database Management* 16, 1 (2005), 33.

[43] Michael Strube and Simone Paolo Ponzetto. 2006. WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI*, Vol. 6. 1419–1424.

[44] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*. 697–706.

[45] Nam Khanh Tran, Tuan Tran, and Claudia Niederée. 2017. Beyond Time: Dynamic Context-Aware Entity Recommendation. In *The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 – June 1, 2017, Proceedings, Part I*, Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig (Eds.). Springer International Publishing, Cham, 353–368. https://doi.org/10.1007/978-3-319-58068-5_22

[46] Howard Turtle, Yatish Hegde, and S Rowe. 2012. Yet another comparison of lucene and indri performance. In *SIGIR 2012 Workshop on Open Source Information Retrieval*.

[47] Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. 2010. Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology (EDBT '10)*. ACM, New York, NY, USA, 697–700. https://doi.org/10.1145/1739041.1739130

[48] Ryen W White and Resa A Roth. 2009. Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services* 1, 1 (2009), 1–98.

[49] Ellery Wulczyn and Dario Taraborelli. 2015. Wikipedia Clickstream. figshare.doi:10.6084/m9.figshare.1305770.

[50] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 283–292.

[51] Lei Zhang, Achim Rettinger, and Ji Zhang. 2016. A Probabilistic Model for Time-Aware Entity Recommendation. In *The Semantic Web – ISWC 2016*, Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil (Eds.). Springer International Publishing, Cham, 598–614.