

Beyond Time: Dynamic Context-Aware Entity Recommendation

Nam Khanh Tran^(✉), Tuan Tran, and Claudia Niederée

L3S Research Center, Leibniz Universität Hannover, Hanover, Germany
{ntran,ttran,niederee}@L3S.de

Abstract. Entities and their relatedness are useful information in various tasks such as entity disambiguation, entity recommendation or search. In many cases, entity relatedness is highly affected by dynamic contexts, which can be reflected in the outcome of different applications. However, the role of context is largely unexplored in existing entity relatedness measures. In this paper, we introduce the notion of contextual entity relatedness, and show its usefulness in the new yet important problem of context-aware entity recommendation. We propose a novel method of computing the contextual relatedness with integrated time and topic models. By exploiting an entity graph and enriching it with an entity embedding method, we show that our proposed relatedness can effectively recommend entities, taking contexts into account. We conduct large-scale experiments on a real-world data set, and the results show considerable improvements of our solution over the states of the art.

Keywords: Contextual entity relatedness · Entity recommendation

1 Introduction

Entities are characterized not only by their intrinsic properties, but also by the manifold relationships between them. Quantifying these entity relationships, which is the idea of entity relatedness [6, 10, 13], is crucial in several tasks such as entity disambiguation [3, 7], contextualization of search results, and improved content analysis [14].

Relationships between entities are not always static. While some relationships are robust and static, e.g. the relationship between a country and its cities, others change frequently, driven by dynamic contexts. In these contexts, time is just one dimension, and alone not sufficient to adequately structure the entity relationship texture. This is illustrated for the entity Brad Pitt in Fig. 1. While time is sufficient to structure the realm of his private relationships, there are other groups of related entities with overlapping timelines, such as the persons he co-acted with in films, which relate to other contexts of his life. Such more fine granular, contextual understanding of the entity relationship texture can be used to refine methods such as entity disambiguation and entity recommendation.

In this paper, we introduce the novel notion of *contextual entity relatedness*, with time and topic as two main ingredients, and show its usefulness in a

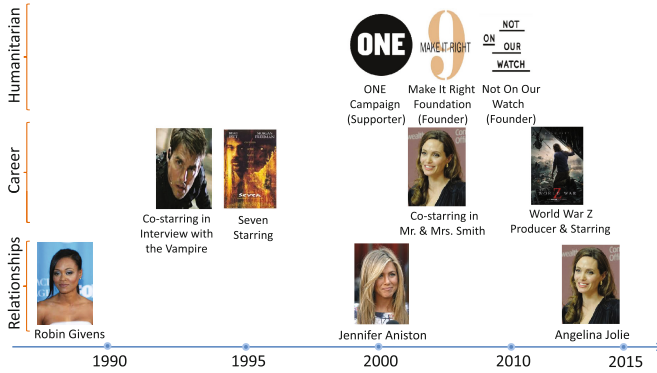


Fig. 1. Related entities with Brad Pitt in different topics and time periods

new yet important problem: Context-aware entity recommendation. We propose to estimate the contextual relatedness using both entity graph extracted from knowledge sources such as Wikipedia, and also to exploit annotated text data using entity embedding methods. Furthermore, while existing work adds temporal aspects into entity relationships [17, 20], we go a step beyond by incorporating topic and proposing to enrich the relationships to form a novel *contextual entity graph*: Each entity relation is enriched with the time span and topics indicating *when* and *under which circumstances* it exists.

The main contributions of this paper are: (1) We introduce the idea of a contextual relatedness of entities (2) we define the problem of context-aware entity recommendation for validating the usefulness of contextual relatedness (3) we propose a novel method for tackling the defined problem based on a statistically sound probabilistic model incorporating temporal and topical context via embedding methods, and (4) we evaluate the context-aware recommendation method with large-scale experiments on a real-world data set. The results of the evaluation show the usefulness of contextual entity relatedness as well as the effectiveness of our recommendation method. compared to other approaches.

2 Background and Problem Definition

2.1 Preliminaries

In this work, we use a very general notion of an entity as “a thing with distinct and independent existence” and assume that each entity has a canonical name and is equipped with a unique identifier. Typically, knowledge sources such as Wikipedia or Freebase are used as reference points for identification.

There are relations between entities. These are represented in different ways such as in the form of hyperlinks in Wikipedia or by a fact in an ontological knowledge base asserting a statement between two entities. Entities and their relationships can be captured in an **entity graph**, where the nodes are entities the edges represent relationships between entities.

An entity can be referred to in a text document (e.g. a news article) in the form of an *entity mention*. In our work, we assume that an **annotated corpus** is given, i.e., an annotated text dataset with well disambiguated entities.¹ Such an annotated corpus can be used to create and enrich the entity graph.

We are interested in the relatedness between entities, which is the association of one entity to another. Such a relatedness is often measured by a normalized score indicating the strength of the association. In our work, these scores depend upon the context and we speak of **contextual relatedness**. For ensuring a wide applicability, we use a simple yet flexible model of context, constituted by two dimensions: Time and Topic. We formalize this concept as follows.

Context. A context c is a tuple (t, s) , where t is a time interval $[t_b, t_e]$ and s is a topic describing the circumstance of the relationship.

Our notion of time is a sequence of discrete time units in a specific granularity, e.g. a day. Time points or ranges of other granularities will be mapped to an interval of this granularity. For example, “2016” is converted to [2016-01-01, 2016-12-31]. For the topic s , we use a textual representation. It can be a single word such as “*movies*”, “*wars*”, or a phrase indicating an information interest such as “*scenes in the thriller movie SEVEN*”.

It is important to note that our contextual relatedness is an asymmetric measure, i.e. given a context c , the relatedness of an entity e_2 to an entity e_1 is different from that of e_1 to e_2 . For example, in the context (2016, “*medals*”), 2016_Summer.Olympics is likely to be the highest related entity for Eri_Tosaka, the Japanese female wrestler² who won her first Olympics gold medals in Rio. The reversed direction is not true, as there are many winners for the total 306 sets of medals in the games.

2.2 Problem Definition

In this work, we aim to study the usefulness of context in entity relatedness. We do this by undertaking a specific recommendation task, namely *context-aware entity recommendation*. In this task, context reflects a user intent or preference in exploring an entity, and contextual relatedness can be used to guide the exploration. Accordingly, by validating the performance of the recommendation task, the effectiveness of contextual relatedness can be evaluated. More specifically, the input of the recommender system is an entity, which the user wants to explore (e.g., Brad_Pitt), and a context consisting of the aspect she is interested in (e.g., (1995-2015, “*awards*”)); the goal is to find the most related entities given the entity and the context of interest. We give the formal definition as follows:

Context-aware Recommendation: Given an entity e_q , a context of interest c_q , an entity graph G , and an annotated corpus D containing annotated and disambiguated entity mentions, find the top- k entities that have the highest relatedness to e_q given the context c_q (contextual relatedness).

¹ Such collections are increasingly available thanks to the advancement in information extraction research. One example is Freebase annotated KBA dataset: <http://trac-kba.org/data/fakba1/>.

² https://en.wikipedia.org/wiki/Eri_Tosaka.

The query (e_q, c_q) is called an entity-context query. The context-aware entity recommendation problem has some assumptions regarding the query setting. First, query entities can have free text representations, but a *text-to-entity* mapping to resolve the canonical entity name is employed. Such a mapping can be the result of using an entity linking system (e.g., [3]). Second, there is also a map from the textual context representation to the time and topic component, for instance “Black Friday 2016 ads” to $([2016-11-25, 2016-11-25], \text{“ads”})$. Third, in the absence of time or topic, they will be replaced by some default place holders. For time, we define two special values b_t and e_t to refer to the earliest and latest days represented in the corpus. For topic, we replace missing values by the token “*” to indicate an arbitrary topic.

3 Approach

This section gives an overview of our method. In essence, we use a probabilistic model to tackle the recommendation task. To estimate the model, we incorporate different graph enrichment methods. These two components are described below.

3.1 Probabilistic Model

We formalize the context-aware entity recommendation task as estimating the probability $P(e|e_q, c_q)$ of each entity e given a entity-context query (e_q, c_q) . The estimation score can be used to output the ranked list of entities. Based on Bayes’ theorem, the probability can be rewritten as follows:

$$P(e|e_q, c_q) = \frac{P(e, e_q, c_q)}{P(e_q, c_q)} \propto P(e, e_q, c_q) \quad (1)$$

where the denominator $P(e_q, c_q)$ can be ignored as it does not change the ranking. The joint probability $P(e, e_q, c_q)$ can be rewritten as:

$$P(e, e_q, c_q) = P(e, e_q, t_q, s_q) = P(e_q)P(t_q|e_q)P(e|e_q, t_q)P(s_q|e, e_q, t_q) \stackrel{\text{rank}}{=} P(e|e_q, t_q)P(s_q|e, e_q, t_q) \quad (2)$$

In (2), we drop $P(e_q)$ and $P(t_q|e_q)$ as they do not influence the ranking. The main problem is then to estimate the two components: $P(e|e_q, t_q)$ (*temporal relatedness* model), and $P(s_q|e, e_q, t_q)$ (the *topical relatedness* model).

3.2 Candidate Entity Identification

The entity graph can be very large, e.g. millions of entities and tens of millions of relationships, thus it is costly to estimate $P(e, e_q, c_q)$ for all entities in the graph. To improve the efficiency, we employ a candidate selection process to identify the promising candidates. Given the query (e_q, c_q) , we extract all entities directly connected to e_q . Other methods can be used in this step; for example entities that co-occur with the target entity in an annotated corpus can be considered as candidate entities. However, in practice, we observe that this strategy covers sufficiently large amount of entities we need to consider.

3.3 Graph Enrichment

To facilitate the estimation methods for Eq. 2 (see Sect. 4 for more details), we propose to enrich the entity graph, i.e. is to equip all entities as well as their relationships with rich information from the knowledge sources and the annotated corpus. This enrichment extends the entity graph into a **contextual entity graph**, where both nodes and edges are contextualized. We describe the enrichment methods below.

Entity Relationship Enrichment. First, we describe how we enrich the graph edges, i.e. the entity relationships. From the annotated corpus, we extract the set of bounded *text snippets* (e.g. a sentence or paragraph)³, in which one or multiple entity mentions to the entities can be found. Then, for each edge (e_i, e_j) , we construct the set of all text snippets annotating both entities e_i and e_j . For each text snippet, we employ a temporal pattern extraction method to extract the time values, and map them to day granularity, or put a placeholder if no values are found. For each successfully constructed time t , we create a context $c = (t, s)$, where s refers to the textual representation of the snippet. As a result, for each edge (e_i, e_j) , we have a set of *relation contexts*, denoted by $C(e_i, e_j)$.

Entity Embedding. To enrich the graph node, i.e. the entity, we propose to learn a continuous vector representation of the entities in the entity graph using a neural network. Our method, *entity embedding*, maps entities to vectors of real numbers so that entities appearing in similar contexts are mapped to vectors close in cosine distance. The vectors can be estimated in a completely unsupervised way by exploiting the distributional semantics hypothesis. Here we extend the Skip-Gram model [9]. The Skip-Gram aims to predict context words given a target word in a sliding window. In our case, we aim to predict context words given a target entity. We train the entities and the words *simultaneously* from the annotated text collection D , using text snippets as the window contexts. Specifically, given a context as a text sequence in which the target entity e appears, i.e., $W = \{w_1, \dots, w_M\}$ where w_i might be either an entity or a word, the objective of the model is to maximize the average log probability

$$\mathcal{L}(W) = \frac{1}{M} \sum_{i=1}^M \log P(w_i | e) \quad (3)$$

in which the prediction probability is defined by using a softmax function

$$P(w_i | e) = \frac{\exp(\vec{w}_i \cdot \vec{e})}{\sum_{w \in W} \exp(\vec{w} \cdot \vec{e})} \quad (4)$$

where \vec{w} and \vec{e} denote the vector representation of w and e respectively. The training example is shown in Fig. 2. The relatedness between two entities e and e_q is then defined as the cosine similarity between their vector representations. In the experiment, we show that the embedding method complements to standard relatedness metrics and help to improve the performance in estimating both models of the contextual relatedness (Eq. 2).

³ In our experiments, we limit to sentences level.

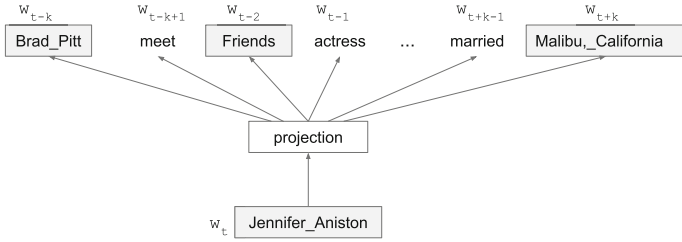


Fig. 2. The training example for the Jennifer Aniston entity

4 Model Parameter Estimation

Our probabilistic model is parameterized by two relatedness models $P(e|e_q, t_q)$ and $P(s_q|e, e_q, t_q)$. In this section, we present in details the estimation of these models based on the contextual entity graph.

4.1 Temporal Relatedness Model

The distribution $P(e|e_q, t_q)$ models the entity relatedness between e and e_q w.r.t t_q . To estimate $P(e|e_q, t_q)$, we take into account both static and dynamic entity relatedness as

$$P(e|e_q, t_q) = \lambda \frac{R_s(e, e_q)}{\sum_{e'} R_s(e', e_q)} + (1 - \lambda) \frac{R_d(e, e_q, t_q)}{\sum_{e'} R_d(e', e_q, t_q)} \tag{5}$$

where $R_s(e, e_q)$ measures the static relatedness between e and e_q , $R_d(e, e_q, t_q)$ measures the dynamic relatedness between e and e_q w.r.t t_q , and λ is a parameter.

Static Relatedness. To measure the static relatedness between entities e and e_q , i.e. $R_s(e, e_q)$, we use the widely adopted method introduced by Milne-Witten et al. using the Wikipedia links [10], and has been effective in various tasks. The Milne-Witten relatedness is measured as:

$$R_s^{MW}(e, e_q) = \frac{\log(\max(|E|, |E_q|)) - \log(|E \cap E_q|)}{\log |V| - \log(\min(|E|, |E_q|))} \tag{6}$$

where E and E_q are the sets of entities that links to e and e_q respectively and V is the set of all entities.

In addition to Milne-Witten, we include the entity embeddings (Sect. 3.3) and define an embedding-based static relatedness measure as the cosine similarity between two corresponding entity vectors:

$$R_s^{Emb}(e, e_q) = \frac{\vec{e} \cdot \vec{e}_q}{\|\vec{e}\| \|\vec{e}_q\|} \tag{7}$$

The two static relatedness measures can be combined in linear fashion to provide the final estimation: $R_s(e, e_q) = R_s^{MW}(e, e_q) + R_s^{Emb}(e, e_q)$.

Dynamic Relatedness. To measure the dynamic relatedness $R_d(e, e_q, t_q)$, we first associate an activation function that captures the importance of an entity e as a function of time: $\alpha_e : T \rightarrow \mathbb{R}$. This function can be estimated by analyzing the edit history of Wikipedia, in which the more edits take place for an article in a certain time interval, the higher the value of activation function. Other kinds of estimators are to analyze longitudinal corpora such as news archives. In this work, our estimation is based on Wikipedia page view statistics. The normalized value of the activation function of an entity α_e is estimated as follows:

$$A_e(t) = \frac{\alpha_e(t) - \mu_{\alpha_e}}{\sigma_{\alpha_e}} \text{ with } \mu_{\alpha_e} = \mathbb{E}[\alpha_e] \text{ and } \sigma_{\alpha_e} = \sqrt{\mathbb{E}[(\alpha_e - \mu_{\alpha_e})^2]} \quad (8)$$

where μ_{α_e} and σ_{α_e} are the mean value and standard deviation of the activation function α_e . To assess whether two entities are temporally related, we compare their activity functions. It happens that many entities exhibit very marked peaks of activity at certain points. These peaks are highly representative for an entity. Therefore, we estimate the dynamic relatedness between entities by measuring a form of temporal peak coherence

$$R_d(e, e_q, t_q) = \sum_{t=t_{q_b}}^{t_{q_e}} \max(\min(A_e(t), A_{e_q}(t)) - \theta, 0) \quad (9)$$

where $t_q = [t_{q_b}, t_{q_e}]$ is the time interval of interest and θ is a threshold parameter that is set as 2.5 here to avoid over-interpreting low and noisy values.

4.2 Topical Relatedness Model

The probability $P(s_q|e, e_q, t_q)$ models the likelihood of observing the text snippet s_q in the relationship between entities e and e_q in the time of interest t_q .

For each context $c_i = (t_i, s_i) \in C(e, e_q)$, let $Sim(s_q, c_i, t_q)$ be the similarity between the text snippet s_q and the context c_i w.r.t the time t_q . The likelihood of observing s_q in the relationship between e and e_q w.r.t t_q is estimated as:

$$P(s_q|e, e_q, t_q) = \frac{1}{|C(e, e_q)|} \sum_{c_i \in C(e, e_q)} Sim(s_q, c_i, t_q) \quad (10)$$

Here we assume the context c_i gives less contribution to the overall relevance of the relation w.r.t the time t_q if its time t_i is distant from t_q , then $Sim(s_q, c_i, t_q)$ is estimated as

$$Sim(s_q, c_i, t_q) = \begin{cases} CS(s_q, s_i) e^{-\beta|t_q - t_i|}, & \text{if } CS(s_q, s_i) \geq \xi \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where ξ is a fixed parameter, β is the decay parameter, $|t_q - t_i|$ is the distance between two time intervals t_q and t_i that is calculated by the distance

between their middle points. The component $CS(s_q, s_i)$ measures the similarity between two text snippets s_q and s_i . We employ two different methods to estimate $CS(s_q, s_i)$, described below.

Language Model. In this method (called *LM-based*), we represent the relation (e, e_q) by a language model, i.e. the distribution over terms taken from text snippets between two entities in the entity graph. Then by assuming the independence between terms in the snippet s_q , we obtain the following estimation

$$CS(s_q, s_i) = \prod_{w \in s_q} P(w|\theta_{s_i})^{n(w, s_q)} \quad (12)$$

where $n(w, s_q)$ is the number of times the term w occurs in s_q , $P(w|\theta_{s_i})$ is the probability of term w within the language model of the snippet s_i which is estimated with Dirichlet smoothing as follows

$$P(w|\theta_{s_i}) = \frac{n(w, s_i) + \mu \cdot P(w)}{\sum_{w'} n(w', s_i) + \mu} \quad (13)$$

where $n(w, s_i)$ is the frequency of w in s_i , $P(w)$ is the collection language model, and μ is the Dirichlet smoothing parameter.

Embedding Model. The second method is an adaptation of the Word Mover’s Distance (WMD) method proposed in [8]. First, we remove all stop words and keep only content words in the text snippets. Then, we define the similarity between two text snippets s_q and s_i using a relaxed version of WMD, where each word in s_q (and s_i) is mapped to its most similar word in s_i (and s_q):

$$CS(s_q, s_i) \propto \frac{1}{2} \left(\frac{\sum_{w \in s_q} \sum_{w' \in s_i} \mathbf{T}_{ww'} \cos(\vec{w}, \vec{w}')}{|s_q|} + \frac{\sum_{w \in s_i} \sum_{w' \in s_q} \mathbf{T}_{ww'} \cos(\vec{w}, \vec{w}')}{|s_i|} \right) \quad (14)$$

where $|s_q|$ and $|s_i|$ are the number words in the text snippets s_q and s_i respectively, $\mathbf{T}_{ww'} = 1$ if $w' = \operatorname{argmax}_{w'} \cos(\vec{w}, \vec{w}')$ or 0 otherwise, $\cos(\vec{w}, \vec{w}')$ is cosine similarity between two vectors. The vector \vec{w} and \vec{w}' are the vector embeddings of the words w and w' , respectively learned from the Entity Embedding method described in Sect. 3.3. We denote this as the *WMD-based* method.

5 Experiment Setup

5.1 Entity Graph Construction

The entity graph we use in the context-aware entity recommendation task is derived from Freebase [4] and Wikipedia.⁴ More specifically, we extract

⁴ English Wikipedia dump version dated March 4, 2015.

Wikipedia articles that overlap with Freebase topics, resulting in 3,866,179 distinct entities, each corresponding to one article. To extract the entity activities for the dynamic temporal relatedness model, we use Wikipedia page view counts⁵ in the time frame 01/01/2012 to 05/31/2016.

We use the text contents of the articles as the annotated corpus D . Note that due to Wikipedia editing guidelines, an article often ignores the subsequent annotations of an entity in the text, if the entity is already annotated before. For example, within the Wikipedia article of entity `Brad_Pitt`, *Angelina Jolie* is mentioned 32 times but only 5 of these mentions are annotated. Hence, we employ a machine learning method [11] to identify more entity mentions. In average, 12 new entity mentions were added to each Wikipedia article.

To extract text snippets for the graph enrichment, we cleaned and parsed the sentences from the contents, resulting in 108 millions sentences in total. We use Stanford Temporal Tagger⁶ to extract temporal patterns from these annotated sentences. For the edges of the entity graphs, we establish the undirected edge (e_1, e_2) if the corresponding Wikipedia article of e_1 or e_2 (after adding new mentions using [11]) contains a hyperlink to the article of the other.

5.2 Automated Queries Construction

We use the recently published Wikipedia clickstream dataset [18] from February 2015 and structural information from Wikipedia for constructing entity-context queries and the Ground Truth.

The clickstream dataset contains about 22 million (referrer, resource) pairs and their respective request count extracted from the request logs of the main namespace of the English Wikipedia. The referrers can be categorized in internal and external traffic; in this work, we only focus on request pairs stemming from internal Wikipedia traffic, i.e., referring page and requested resource are both Wikipedia pages from the main namespace.

Wikipedia articles are collaboratively and iteratively organised in sections and paragraphs, such that each section is concerned with particular aspects or contexts of the entity profile [5]. Each entity mentions within these sections are therefore highly relevant to the source entity in the respective context.

Based on these observations, we propose an automated entity-context query construction using the following heuristics: (i) For each pair of source and target entities, we first extract the section heading where the target entity is mentioned in the source page (ii) The source entity is then used as query entity and the extracted heading is used as context to create a entity-context query; here we filter out noisy headings such as “*further reading*”, “*see also*”. (iii) We only keep queries for which at least 5 entities are clicked in the clickstream dataset.

To construct the query time, we use the publication time of clickstream dataset, which is February 2015, and convert it to [2015-02-01,2015-02-28].

⁵ <https://dumps.wikimedia.org/other/pagecounts-ez/>.

⁶ <http://nlp.stanford.edu/software/sutime.shtml>.

Table 1. Example of entity-context queries and related entities with the number of clicks extracted from the clickstream dataset

Entity	Context	Related entities
Brad Pitt	Humanitarian and political causes	University of Missouri (101), John Kerry (80), Barack Obama (26)...
Brad Pitt	Career	Fury (2014 film) (1772), Mr. & Mrs. Smith (2005 film) (973), Legends of the Fall (893)...
Brad Pitt	Personal life	Angelina Jolie (16564), Jennifer Aniston (11306), Gwyneth Paltrow (3383)...
Brad Pitt	In the media	Supercouple (798), People (magazine) (126)...

Table 1 presents example queries created for the entity `Brad_Pitt`. In total, we have 219,844 entity-context queries. To accommodate the impact of time in the queries, we define the *ratio of views*, denoted by r , which is the ratio between the number of times the entity was clicked in February 2015 and in January 2015. The intuition is that if r is very high, the corresponding query entities and topics might have some underlying information interests emerging in February 2015 (for instance, the release of a new movie, etc.). We divide our query set into 4 subsets based on different value ranges of r (Table 2).

Table 2. The different set of queries Q_r with varying ratios of interest

Query set	$Q_{r>0}$	$Q_{r>1}$	$Q_{r>5}$	$Q_{r>10}$
Number of queries	219,844	69,489	1,263	493

Ground Truth. For each query in the query set, we establish the ground truth through the click information available in the clickstream dataset. Existing work suggests that the Wikipedia viewing behaviour can be used as a good proxy of entity relevance to current user interest [12, 15]. Transferring this idea to navigational traffic within Wikipedia networks (as they are reflected in the click streams), we can consider an increased navigation between two entities as a signal for the importance of the relationship between the corresponding source and the target entities.

Thus, given an entity-context query, the larger number of clicks a candidate entity gets, the higher related the entity is. Based on this, for each query we take the most clicked entity as the relevant entity, and measure how good recommendation approaches rank the entity using *MRR* metric. In addition, we extract the top-5 clicked entities for each query to measure the recall. We publish our code and data to encourage future similar research.⁷

⁷ <http://www.l3s.de/~ntran/dycer.html>.

Evaluation Metrics. To measure the performance of different approaches, we use two evaluation metrics. The first metric is *mean reciprocal rank* (**MRR**) which is computed as

$$MRR = \frac{1}{|Q_{test}|} \sum_{i=1}^{|Q_{test}|} \frac{1}{rank(e_{q_i})} \quad (15)$$

where $|Q_{test}|$ is the number of queries, and $rank(e_{q_i})$ represents the rank of the ground truth entity e_{q_i} in the results for the query q_i . Notice that a larger MRR indicates better performance.

We also use **recall** at rank k ($R@k$) as another evaluation metric. $R@k$ is measured as the ratio of the retrieved and relevant entities up to rank k over the total number of relevant results. The larger $R@k$ indicates better performance.

5.3 Baselines

We implemented several baselines to compare to our methods on the task. The first group of baselines are static methods using an ad hoc ranking function without considering the given context. We consider the baselines that only use Milne-Witten or entity embeddings-based relatedness, and the combination. We denote these static methods as **Static**_{*mw*}, **Static**_{*emb*}, and **Static**_{*mw&emb*}.

The second group of baselines are time-aware methods which are similar to our probabilistic model but without taking into account the search topic s_q . We reimplemented the approach proposed by [20] and extended it by combining the entity embedding and link based similarities to integrate into the model. We denote these time-aware methods as **Temp**_{*mw*} [20] and **Temp**_{*mw&emb*}.

Finally, we denote our methods as **Dycer**_{*lm*} and **Dycer**_{*wmd*} where **Dycer**_{*lm*} uses the LM-based method and **Dycer**_{*wmd*} uses the WMD-based method for estimating the similarity between text snippets.

Parameter Settings. We empirically set the similarity threshold ξ to 0.35, and the decay parameter β to 0.5. The Dirichlet smoothing parameter is fixed to 2000, and the parameter λ is set to 0.3 by default and will be discussed in detail in the experiments.

6 Results and Discussion

Figure 3 presents a detailed comparison between the MRR for the different methods. The proposed methods outperform the baselines on all query sets. In addition, when increasing the ratio of views r , our method progressively improves, with its highest score $MRR = 0.282$ on the query set $Q_{r>10}$. In contrast, the performance of the static methods is not changed much and around $MRR = 0.145$. This conforms the effectiveness of our model in capturing the dynamic contexts. Even without context, our relatedness model (*Static*_{*mw&emb*}) already performs better compared to the *Static*_{*mw*} and *Static*_{*emb*} methods. Interestingly, the time-aware methods gain comparable, even worse results compared

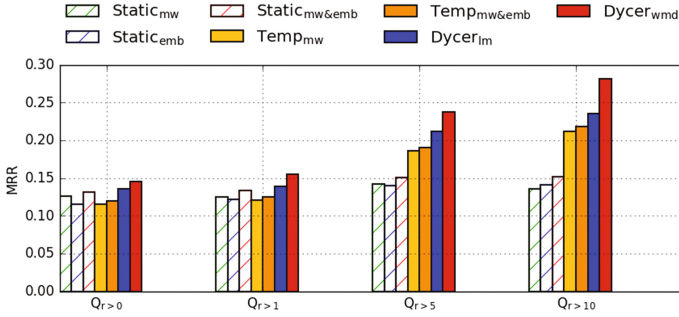


Fig. 3. Performance of the different approaches on the different query sets

to the static methods on the query sets $Q_{r>0}$ and $Q_{r>1}$, however they obtain significantly better *MRR* scores on the query sets $Q_{r>5}$ and $Q_{r>10}$. This can be explained by the fact that the entities in $Q_{r>5}$ and $Q_{r>10}$ are more sensitive to time because of high user interests. Furthermore, the adapted implementation $Temp_{mw\&emb}$ outperforms the original method $Temp_{mw}$, which again indicates the effectiveness of the combination of the embedding-based and link-based methods. The best overall performing approach is the WMD-based method $Dycer_{wmd}$. The method performs better than the LM-based method $Dycer_{lm}$, which is due to the fact that the WMD-based method takes into account the semantic meaning of words using word embeddings for the textual similarity estimation, while the LM-based method purely uses the surface form of words.

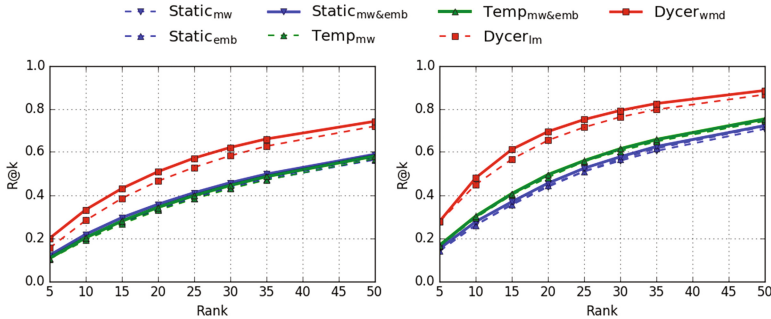


Fig. 4. $R@k$ for the different entity recommendation approaches under comparison. (Left) All queries $Q_{r>0}$. (Right) Queries with high ratios $Q_{r>5}$

Next, we analyse the recall at rank k ($R@k$) as quality criteria. The results of $R@k$ with varying k for different methods are shown in Fig. 4. We compute the performance of methods on the different query sets $Q_{r>0}$ and $Q_{r>5}$. Figure 4 shows that the proposed methods outperform the baselines on both sets of queries. On the first query set $Q_{r>0}$ the WMD-based method $Dycer_{wmd}$ gains

7.9%, 10.5%, 15.2%, and 14.3% improvements compared to the static method $Static_{mw\&emb}$, and 9.3%, 13.0%, 16.6% and 17.6% improvements compared to the time-aware method $Temp_{mw\&emb}$ when the rank k is 5, 10, 20, and 30 respectively. On the query set $Q_{r>5}$, it even obtains much better improvements. In addition, similar to our findings for MRR , the time-aware methods achieve comparable results compared to the static methods overall, but perform considerably better on the query set with the high ratio of views $Q_{r>5}$.

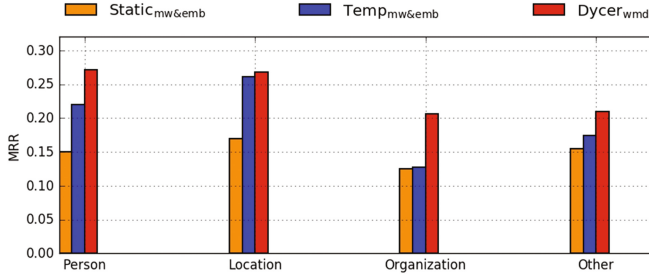


Fig. 5. MRR of relevant entity for different query entity types in $Q_{r>5}$ and for different approaches (note, we show the results for the best method in each group)

In addition, we also compare the performance in different query types, as for each type, users often have different intents and expectations. Figure 5 shows the comparison in terms of MRR for four groups of high-level types. It can be seen that the performance differences vary quite noticeably in different type groups. Nevertheless, in all cases, the highest result is achieved by the $Dycer_{wmd}$ approach. Interestingly, for the type “*Person*” and “*Location*” the $Temp_{mw\&wmd}$ approach gains large improvements compared to the static method. One possible explanation for this is that the “*Person*” and “*Location*” entities usually involve in events which highly relate to time. Consequently, taking time into account helps improving the performance. In the case of “*Organization*”, the time-aware method does not show any improvement compared to the $Static_{mw\&emb}$ method whereas the WMD-base method still obtains a huge improvement. It demonstrates the usefulness of contextual information for the task.

Table 3 shows the impact of λ on the performance of the time-aware and the proposed method using the query set $Q_{r>5}$. The $\lambda = 0.3$ yields the best results on average using both methods, which is then used as in our experiments.

Table 3. MRR of relevant entity using the query set $Q_{r>5}$ for different λ (with the best results in bold)

Method	$\lambda = 0$	$\lambda = .1$	$\lambda = .2$	$\lambda = .3$	$\lambda = .4$	$\lambda = .5$	$\lambda = .6$	$\lambda = .7$	$\lambda = .8$	$\lambda = .9$	$\lambda = 1$
$Temp_{mw\&emb}$	0.1700	0.1894	0.1902	0.1914	0.1913	0.1913	0.1903	0.1906	0.1888	0.1898	0.1511
$Dycer_{wmd}$	0.2153	0.2371	0.2372	0.2372	0.2359	0.2366	0.2365	0.2360	0.2361	0.2363	0.1828

Discussions. While we use Wikipedia for building the model in the experiments, the proposed approach can also use other knowledge bases (e.g. Freebase) to construct the entity graph, and any text collections (e.g. news archives, web archives) can also be used to enrich the entity graph. Our choice of using Wikipedia is driven by the availability of rich and high-quality meta-data in the collection, which enables us to focus on the effectiveness of the models. In addition, we focus on frequent entities in our experiments, however the proposed method leverages both link structures and the textual representation from the document collection to estimate the entity relatedness; thus we believe that it can achieve good performance with the long-tail entities, as been shown in existing approaches [6]. These evaluations are left for future work.

7 Related Work

Estimation of entity semantic relatedness is an important task in various semantic and NLP applications, and has been extensively studied in literature [10, 13]. Strube and Ponzetto [13] proposed using Wikipedia link structures and the hierarchy of Wikipedia categories to provide a light-weight related estimation. Milne and Witten [10] followed a similar approach, and carefully designed the relatedness measure based on Wikipedia incoming links, inspired by the Google distance metric. These methods are close to our work in the sense that we also combine various similarity measures, but do so in an advanced probabilistic model, taking into account context information. Hence, while the aforementioned works are static, our proposed measure is context-aware and dynamic to time.

One main issue with relatedness measures based on link structures is that they perform poorly for long-tail entities with little or no connections. Hoffart et al. [6] (KORE) addressed this issue by extracting key phrases from surrounding texts of entity mentions, and incorporate the overlaps of such key phrases between two entities. In our work, we also use the text surrounding of entity mentions. However, in contrast to KORE that uses these texts to enrich the entities, we use the texts to enrich the relations between entities, and in this regard, can contextualize the relatedness directly. In addition, KORE is still a static quantity, while our measure is fully dynamic to time and context.

Several approaches have been proposed to add temporal dimension to entity semantic relationships [16, 17]. Wang et al. [17] extracted temporal information for entities with focus on infobox, categories and events. Tuan et al. [16] also extracted information from infobox and categories, but defined a comprehensive model comprising time, location and topic. However, these studies are limited to predefined types of relations, and cannot be easily extended to address the semantic relatedness. Recently, Zhang et al. [20] incorporated various correlation metrics to complement the semantic relatedness, proposed a new metric that is sensitive to time. We extend this work, but incorporate time and topic in an consistent context model, and also introduce the entity embedding method.

From the application perspective, entity recommendation is one of the directed applications of entity semantic relatedness. It assumes the input entities

encode some user activities or information needs, and suggests a list of entities, normally ordered, that are most relevant. Blanco et al. [2] introduced Spark that links a user search query to an entity in a knowledge base and suggests a ranked list of related entities for further exploration. Similarly, [1, 19] proposed personalized entity recommendation which uses several features extracted from user click logs. Our work is distinguished from this work in that we take into account context as an additional information need, not just input entities.

8 Conclusions

In this work, we have presented the concept of contextual entity relatedness and proposed effective methods for using it in the task of entity recommendation. We have shown the usefulness of the contextual information for this task as well as the effectiveness of our method.

Our work leaves ample space for further investigations in contextual entity relatedness. One research direction is the investigation of further more fine granular context dimensions. In addition, we have planned to look into more depth the evolution of entity relationship and to exploit contextual information for improving performance of other tasks such as entity disambiguation.

Acknowledgments. This work was partially funded by the German Federal Ministry of Education and Research (BMBF) for the project eLabour (01UG1512C).

References

1. Bi, B., Ma, H., Hsu, B.J.P., Chu, W., Wang, K., Cho, J.: Learning to recommend related entities to search users. In: WSDM (2015)
2. 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.) ISWC 2013. LNCS, vol. 8219, pp. 33–48. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-41338-4_3](https://doi.org/10.1007/978-3-642-41338-4_3)
3. Blanco, R., Ottaviano, G., Meij, E.: Fast and space-efficient entity linking in queries. In: WSDM (2015)
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD (2008)
5. Fetahu, B., Markert, K., Anand, A.: Automated news suggestions for populating wikipedia entity pages. In: CIKM (2015)
6. Hoffart, J., Seufert, S., Nguyen, D.B., Theobald, M., Weikum, G.: KORE: keyphrase overlap relatedness for entity disambiguation. In: CIKM (2012)
7. 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 (2011)
8. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: ICML (2015)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR (2013)

10. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: AAAI (2008)
11. Noraset, T., Bhagavatula, C., Downey, D.: Adding high-precision links to wikipedia. In: EMNLP (2014)
12. Ratkiewicz, J., Flammini, A., Menczer, F.: Traffic in social media i: paths through information networks. In: SocialCom (2010)
13. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: AAAI (2006)
14. Tran, N.K., Ceroni, A., Kanhabua, N., Niederée, C.: Supporting interpretations of forgotten stories by time-aware re-contextualization. In: WSDM (2015)
15. Tran, T.A., Niederee, C., Kanhabua, N., Gadiraju, U., Anand, A.: Balancing novelty and salience: adaptive learning to rank entities for timeline summarization of high-impact events. In: CIKM (2015)
16. Tuan, T.A., Elbassuoni, S., Preda, N., Weikum, G.: CATE: context-aware timeline for entity illustration. In: WWW (2011)
17. Wang, Y., Zhu, M., Qu, L., Spaniol, M., Weikum, G.: Timely YAGO: harvesting, querying, and visualizing temporal knowledge from wikipedia. In: EDBT (2010)
18. Wulczyn, E., Taraborelli, D.: Wikipedia clickstream (2015)
19. Yu, X., Ma, H., Hsu, B.J.P., Han, J.: On building entity recommender systems using user click log and freebase knowledge. In: WSDM (2014)
20. Zhang, L., Rettinger, A., Zhang, J.: A probabilistic model for time-aware entity recommendation. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) ISWC 2016. LNCS, vol. 9981, pp. 598–614. Springer, Cham (2016). doi:[10.1007/978-3-319-46523-4_36](https://doi.org/10.1007/978-3-319-46523-4_36)