Ranking Associative Entities in Knowledge Graph by Graphical Modeling of Frequent Patterns

Jie Li, Kun Yue(✉), Liang Duan, and Jianyu Li

School of Information Science and Engineering, Yunnan University, Kunming, China
{jiel,jylee}@mail.ynu.edu.cn, {kyue,duanl}@ynu.edu.cn

Abstract. Ranking associative entities in Knowledge Graph (KG) is critical for entity-oriented tasks like entity recommendation and associative inference. Existing methods benefit from explicit linkages in KG w.r.t. exactly two query entities via the closely appearing co-occurrences. Given a query including one or more entities in KG, it is necessary to obtain the implicit associative entities and uncover the strength of associations from data. To this end, we leverage KG with Web resources and propose an approach to ranking associative entities based on frequent pattern mining and graph embedding. First, we construct an entity dependency graph from the frequent patterns of entities generated from both KG and Web resources. Thus, the existence and strength of associations between entities could be depicted effectively in a holistic way. Second, we embed the dependency graph into a lower-dimensional space and consequently fulfill entity ranking on the embedding. Finally, we conduct an extensive experimental study on real-life datasets, and verify the effectiveness of our proposed approach compared to competitive baselines.

Keywords: Knowledge graph · Associative entity · Association ranking · Frequent entity · Graph embedding

1 Introduction

Many entity-oriented applications, like entity alignment [4], entity recommendation [20] and entity associations inference [16], benefit from the results of top-ranked associative entities in knowledge graph (KG). The task of ranking associative entities (a.k.a. association ranking) is to sort candidate entities w.r.t. a query including one or more given entities in KG. For example, {*1. Microsoft*; *2. COVID-19*; *3. Windows*} is a ranking list of candidate entities w.r.t. the query entity *Bill Gates* sorted by their association strength.

It is straightforward to represent associative entities based on the triple-structured data of KG. One feasible solution to ranking associative entities is

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12681, pp. 224–239, 2021.
https://doi.org/10.1007/978-3-030-73194-6_16

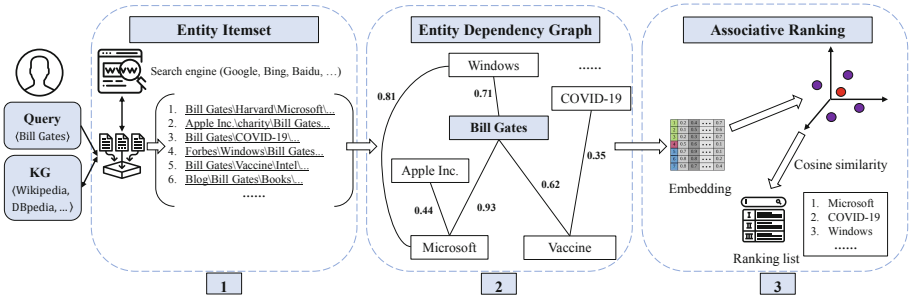


Fig. 1. Overview of EDGM.

based on the semantic associations (a.k.a. relatedness) between words or entities [19] upon the prerequisite that frequently occurring entities are regarded to be highly associated. However, only the frequencies of two closely appearing entities are considered, while the highly associated entities do not necessarily co-occur significantly in the neighboring context. The underlying local co-occurrence principle leads to limited coverage and precision. Recently, multiple association features between words, concepts, and entities are combined to construct an association network [9,15] to improve the relatedness measurement. However, these models could not be well learned in an unsupervised manner. Meanwhile, these methods focus on measuring the semantic association between exactly two entities within KG. It will be more scalable if any number of query entities are allowed and multiple Web resources could be introduced.

By using linked Web resources, explicit associations could be found easily to enhance Web applications like search engines [21]. However, implicit associations between entities show usefulness in many domains including national security and biomedical research [5]. For example, it is necessary to identify the importance of implicit associations such as common preferences and similar behaviors in social networks. Potential connections between a group of users may contribute to suspect search. Thus, additional Web resources outside KG are incorporated to improve the ranking results [14,26]. Figure 1 illustrates an example of ranking associative entities based on both KG and Web resources. As shown in part 2, *Bill Gates* and *Vaccine* are associative with the strength of 0.62, which corresponds to the news topic of *"Bill gates pledges $1.6 billion to vaccine research against COVID-19"*. That is, even though *Bill Gates* and *Vaccine* are not directly linked in KG, there is still a strong association between them. By introducing retrieved results of Web pages w.r.t. *Bill Gates* in part 1, it is available to uncover these kinds of implicit associations.

Thus, we consider refining the association features by incorporating KG and Web resources to fulfill the task of ranking associative entities, in which we will have to solve the following 2 questions:

(1) How to find the associative entities w.r.t. one or more query entities in KG using association features from both KG and Web resources?
(2) How to measure the strength of associations between entities by holistically aggregating multiple co-occurrences?

In this paper, as shown in Fig. 1, we propose an Entity Dependency Graph Model (EDGM) to rank associative entities by graph embedding. In our EDGM, we use association features from both KG and Web resources based on associative Wikipedia articles and contents of Web pages w.r.t. the query entities. The associations especially for up-to-date situations could be determined by the rapidly changing or generated Web resources or user behavioral records that we regard as the *transactions* in frequent pattern mining [1]. By this way, we propose a method to bridge the gap between frequent pattern mining and graphical model. Upon the graph structure of frequent entities, we aggregate frequencies of both single entities and the co-occurrences of entities to evaluate the associations quantitatively. To obtain highly represented embedding of associative entities and fulfill effective ranking, we adopt a BFS-biased random walk sampling mechanism based on node2vec [10]. This enables our EDGM to better measure the strength of associations by capturing neighboring and co-occurring features accurately. The contributions of this paper are as follows:

First, as illustrated in part 1 of Fig. 1, we generate an entity itemset containing sequences of candidate entities to the query from both KG and Web resources. By incorporating the extracted Web resources, it is practical to integrate various statistics of entities. Then, we adopt the frequent pattern mining algorithm on the entity itemset to build an undirected weighted graph, where each node represents an entity, and each edge represents the associations between entities. By an unsupervised manner, co-occurrence associations w.r.t. one or more query entities could be discovered.

Second, to improve the effectiveness of ranking associative entities, we measure the weight of each edge on EDG, which could present the strength of associations by refining both the informativeness and specificness of co-occurrences simultaneously. To fulfill entity ranking for each associative candidate to the given query, we use graph embedding to transform the nodes on the weighted graph into a low-dimensional space and then rank the candidate entities based on the similarity between node embeddings.

Finally, we conduct extensive experiments on two real-life datasets to evaluate the effectiveness of our EDGM. Experimental results illustrate that our approach outperforms some state-of-the-art competitors in ranking associative entities.

The rest of this paper is organized as follows: Sect. 2 introduces related work and preliminaries. Section 3 presents our methods for learning EDG and ranking associative entities. Section 4 shows experiments and performance studies. Section 5 concludes and discusses future work.

2 Related Work and Preliminaries

In this section, we review related work, followed by giving necessary definitions and formulating the problem.

2.1 Related Work

Most research efforts for ranking associative entities could be divided into 3 categories: entity relatedness ranking, association ranking of KG, and entity ranking by graph embedding.

Entity Relatedness Ranking. Entity relatedness ranking optimizes the partial order of the associative entities into desired positions upon semantic relatedness [20]. For measuring relatedness between exactly two entities, text-based methods [2,8] build high-dimensional weighted vectors to represent words and Wikipedia concepts. Other graph-based approaches [27] adopt the link structure of Wikipedia to obtain the distance of entities. These methods are insufficient to uncover more profound co-occurrences with only text semantics or graphical structural relatedness. Better results could be achieved by integrating existing methods through designing comprehensive frameworks [25]. To further leverage more types of co-occurrences in KG, network-based methods [9,15] specify associations among words and concepts in a supervised manner upon well-generated datasets from psychological studies.

Association Ranking of KG. Techniques for ranking associations between two or more entities are developed with the emergence of graph-structured Web resources, which could be divided into data-centric and user-centric. Data-centric techniques mainly use various statistical information of entities, and user-centric techniques focus on user preference. Typically, the associations are regarded as paths connecting two or more entities in KG [7]. Simple associations could be obtained directly by triple-linked data from KG, but implicit associations are more preferred in some domains [5]. To search and rank implicit associations, the frequent pattern mining algorithm has been proved to be efficient and effective [6]. By counting the frequency of canonical codes uniquely representing entity patterns, associations could be ranked upon the edit distance between graph structures.

Entity Ranking by Graph Embedding. Graph embedding techniques like DeepWalk [23] are effective for association analysis in graphical structures [3], in which low-dimensional representations of the nodes with neighboring and co-occurrence relations are learned. Zhang et al. [29] propose a graph embedding-based neural ranking framework to overcome the query-entity sparsity problem by integrating features in click-graph data. On heterogeneous information networks, recent studies for proximity search [18] learn graph embedding models to rank associative nodes by given semantic relations. These techniques are based on user intent with a certain amount of behavior preference labels. Differently, we choose node2vec [10] to embed the associations between entities, since the co-occurrences on EDG, together with their strength, could be expressed by using the biased and dynamic random walk.

2.2 Definitions and Problem Formulation

Firstly, the symbols and notations are given in Table 1. Then, we define several concepts as the basis of later discussions.

Table 1. Notations.

Notation	Description
$D_{\vec{q}}$	Associative datasets w.r.t. query $\vec{q}$
$\Psi(\vec{q})$	Entity itemset w.r.t. $\vec{q}$
Υ	Set of all 1-frequent entities
A_x	A maximal set of frequent entities
$G_E = (V, E, W)$	Entity dependency graph with nodes V, edges E and weights W
$\overline{v_i v_j}$	Edge between 1-frequent entities v_i and v_j
$H^{\lvert V\rvert \times d}$	Representation space of EDG with the dimension of $\lvert V\rvert \times d$
$L(\vec{q})$	Ranking list of associative entities w.r.t. $\vec{q}$

To obtain $D_{\vec{q}}$ from both the KG and the Web, associative data like Web pages and Wikipedia articles w.r.t. the query entities $\vec{q}$ could be retrieved and collected by search engines.

Definition 1. *A knowledge graph is denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ represents a finite set of nodes indicating entities, and $\mathcal{R}$ is a set of directed edges representing relations between entities.*

Sequences of associative entities could be generated based on items in $D_{\vec{q}}$ and named entities of $\mathcal{G}$. The definition of $\Psi(\vec{q})$ is as following:

Definition 2. *Let $\psi = \{e_1, e_2, ..., e_M\}$ be a sequence of entities, where $e_i \in \mathcal{E}$ and $\psi \in \Psi(\vec{q})$. Each ψ is corresponding to an item in $D_{\vec{q}}$.*

Based on the idea of frequent pattern mining [1], $\Psi(\vec{q})$ could be regarded as the *transactions* of $D_{\vec{q}}$. Next, we define the *set of frequent entities*.

Definition 3. *$v (v \in \mathcal{E})$ is called a 1-frequent entity if $p(v) \geq \sigma$, where $p(v)$ is the* support *of v (i.e., the proportion of sequences in $\Psi(\vec{q})$ containing v) and σ is the* threshold of minimal-support. *The set of all 1-frequent entities is denoted as Υ.*

Definition 4. *A set of frequent entities $A_x \subset \Upsilon$ is called* maximal, *if there are no other super-sets A_y in $\mathbb{A}$ satisfying $A_x \subset A_y$, where $\mathbb{A} = \{A_1, ..., A_m\}$ includes all the sets of frequent entities.*

Following, we define the entity dependency graph (EDG) to describe the existence and strength of associations between entities.

Definition 5. *An EDG is an undirected weighted graph, denoted as $G_E = (V, E, W)$. V is the set of nodes, and $V \subset \Upsilon$. Each edge $\overline{v_i v_j} \in E$ $(v_i, v_j \in V, i \neq j)$ indicates the co-occurrence association between v_i and v_j. Each $w_{ij} \in W$ represents the weight of $\overline{v_i v_j}$.*

Problem Formulation. Given the query $\vec{q}$, we first extract its itemset $\Psi(\vec{q})$ from $D_{\vec{q}}$ as the input to construct EDG. For each node in EDG, the representation space H is learned as:

$$f : V \longrightarrow H^{|V| \times d} \tag{1}$$

Upon the matrix $H^{|V|\times d}$, we measure the strength of associations between each candidate entity and $\vec{q}$ from a global perspective, and output the ranking list of candidate entities $L(\vec{q})$ w.r.t. $\vec{q}$.

3 Methodology

In this section, we introduce the approach to ranking associative entities by our EDGM. First, the structure of EDG is learned by mining frequent patterns from the *transactions* of both KG and Web resources, and then the weights of edges on EDG are measured based on an extension principle of co-occurrences. Finally, the ranking process is implemented by graph embedding.

For the given KG $\mathcal{G}$ and query $\vec{q}$, the sequences of entities recognized from KG are *transactions* of $D_{\vec{q}}$, for which the entity itemset $\Psi(\vec{q})$ is generated from $D_{\vec{q}}$ (e.g., Wikipedia articles and Web pages retrieved w.r.t. $\vec{q}$) by entity linking. Then, by learning the graphical structure and measuring the weights of edges, the EDG $G_E = (V, E, W)$ is constructed to depict the associations between frequently co-occurring entities in a holistic way.

3.1 Structure Learning

Learning the structure of G_E aims to determine the set of nodes V and the set of edges E. The nodes in V are generated by mining frequent entities in $\Psi(\vec{q})$, and the edges in E depend on the test of conditional independence [17] between frequent entities.

To achieve a high recall in line with the inherence of co-occurrence between entities, the node set V should contain the candidate entities related to the query as many as possible. Given $\Upsilon = \{v_1, v_2, ..., v_n\}$ as a set of 1-frequent entities in $D_{\vec{q}}$, we generate V from Υ by neglecting the entities whose support values are less than the threshold σ according to the probability cut defined as follows:

$$p_\sigma(I) = \begin{cases} 0 & p(I) < \sigma \\ p(I) & p(I) \geq \sigma \end{cases} \tag{2}$$

As is known that only frequent entities are concerned when computing $p(I)$ by the classic Apriori algorithm [11]. If I is a set of frequent entities, then all the non-empty subsets of I must also be frequent. If there is no set of frequent entities J in such Υ that $I \subset J$, we call I is the maximal. To include the entities concerning all co-occurrences, we adopt the entities in the maximal set of frequent entities as nodes in V.

To determine the edges among the nodes in V, we first generate completely connected subgraphs over each maximal set of frequent entities. According to the conclusion in [17], the associations between frequent entities imply probabilistic conditional independences. Thus, two entities in the set of frequent entities are not connected in G_E by an edge if they are conditionally independent. By testing

conditional independence, the graphical topology of frequent entities could be obtained.

The conditional independence between entities is closely related to the frequent set, to which the entities belong. Let I, J and K be three disjoint subsets of Υ. We use $< I|K|J >$ to denote that "I is independent of J given K", namely $p(I \cap J|K) = p(I|K)p(J|K)$. By focusing on the conditional independence relations between frequent entities, we analyze possible associations between them.

Specifically, let I and J $(I, J \subseteq \Upsilon)$ be two different sets of maximal frequent entities and $I \cap J = K$. We consider the following three cases. For two entities in different sets with intersections, an edge is added between these two entities to reflect their mutual dependency. Two entities in different sets without overlap are unconnected. Two entities are also unconnected if they are already in one set and do not co-occur in any other sets.

Case 1. If there exist 1-frequent entities $v_a \in I$ and $v_b \in J$ such that the entity set $\{v_a, v_b\}$ is non-frequent, then $< v_a|K|v_b >$ is false, denoted as $\overline{< v_a|K|v_b >}$. That is, v_a and v_b are associative and there is an edge $\overline{v_a v_b}$.

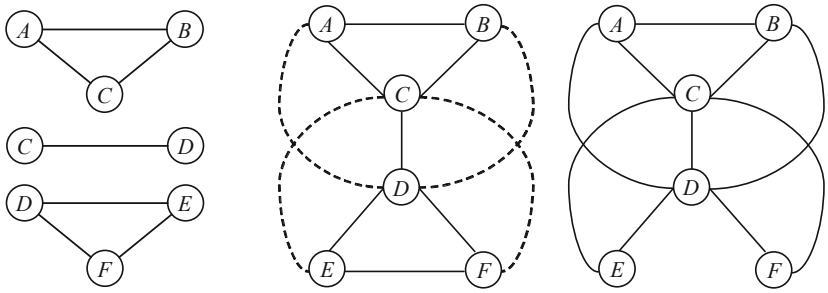
Case 2. If $\{v_a, v_b\}$ is non-frequent for all $v_a \in I$ and $v_b \in J$ when $K = \varnothing$, then $< v_a|\Upsilon - v_a - v_b|v_b >$ is true. In other words, if there is no any frequent entity v_c such that $\{v_a, v_c\}$ and $\{v_b, v_c\}$ are frequent when $\{v_a, v_b\}$ is non-frequent, v_a and v_b are independent and there is no edge between them.

Case 3. Suppose $< v_a|I - v_a - v_c|v_c >$ is true, where $v_a, v_c \in I$. If there is no such J that $v_a, v_c \in J$, then $< v_a|\Upsilon - v_a - v_c|v_c >$ is true. That is, two conditionally independent entities v_a and v_c do not share an edge if they co-occur only in one maximal set of frequent entities.

Algorithm 1. Structure learning of EDG

Input: Υ; $\mathbb{A} = \{A_1, ..., A_m\}$, where each $A_x \in \mathbb{A}$, $A_x = \{v_{xy} | v_{xy} \in \Upsilon, 1 \leq y \leq n\} (x \in [1, m])$
Output: V, the set of nodes in G_E; E, the set of edges in G_E

$V \leftarrow \Upsilon$, $E \leftarrow \{\}$, $G_{\mathbb{A}} \leftarrow \{\}$
for each $A_x \in \mathbb{A}$ **do**
  Generate $G_{A_x}(V_{A_x}, E_{A_x})$ // Join each pair of distinct entities in A_x
  $G_{\mathbb{A}} \leftarrow G_{\mathbb{A}} \cup G_{A_x}$ // G_{A_x} is the complete graph of A_x and $G_{\mathbb{A}}$ is the set of G_{A_x}
end for
for each pair $(A_x, A_y) \in \mathbb{A} \times \mathbb{A}$ **do** // Case 1
  if $A_x \cap A_y \neq \varnothing$ **then**
    for each edge $\overline{v_{xs} v_{yt}}$ **do** // $v_{xs} \in A_x - A_y$ and $v_{yt} \in A_y - A_x$
      $E \leftarrow E \cup \overline{v_{xs} v_{yt}}$ // Add $\overline{v_{xs} v_{yt}}$ to the set of edges
    end for
  end if
end for
for each $G_{A_x} \in G_{\mathbb{A}}$ **do**
  for each edge $\overline{v_{xs} v_{xt}} \in G_{A_x}$ **do**
    if $< v_{xs}|A_x - v_{xs} - v_{xt}|v_{xt} >$ **then** // Case 2
      $E \leftarrow E - \overline{v_{xs} v_{xt}}$
    end if
    for each $A_y \in \mathbb{A} - A_x$ **do** // Case 3
      if $< v_{xs}|A_y - v_{xs} - v_{xt}|v_{xt} >$ or $v_{xs} \notin A_y$ or $v_{xt} \notin A_y$ **then**
        $E \leftarrow E - \overline{v_{xs} v_{xt}}$
      end if
    end for
  end for
end for
return V, E



(a) Three undirected complete subgraphs (b) Undirected graph generated by Algorithm 1 (c) Results EDG over Υ

Fig. 2. A running example of Algorithm 1.

Next, we illustrate the execution of Algorithm 1 by the following example. Given $\Upsilon = \{A, B, C, D, E, F\}$ as the set of 1-frequent entities. $\{A, B, C\}$, $\{C, D\}$, and $\{D, E, F\}$ are three maximal sets of frequent entities over Υ.

Firstly, we add edges for entities within one maximal set of frequent entities respectively in Fig. 2(a) to generate three undirected complete subgraphs according to Case 1. Secondly, we add undirected edges $\overline{AD}$, $\overline{CE}$, $\overline{BD}$ and $\overline{CF}$ shown by dotted lines in Fig. 2(b) to represent the possible associations. According to Case 2, following edges should not exist: $\overline{AE}, \overline{AF}, \overline{BE}, \overline{BF}$. Finally, suppose that the conditional independence tests show $< E|D|F >$ and $< E|C|F >$ are true. Then, according to Case 3, $\overline{EF}$ will be deleted. The actual structure of EDG is shown in Fig. 3(c).

Step 2 in Algorithm 1 could be done in $O(|A_1|^2 + ... + |A_m|^2)$ time, and does not exceed $O(m \times n^2)$, where $|A_x|$ is the number of entities in A_x and $|A_x| \leq n (1 \leq x \leq m)$. Step 6 could be done in $O(m \times n^2)$ time at most. Step 13 could be achieved in $O(|A_1|^2 + ... + |A_m|^2)$ time and no larger than $O(m \times n^2)$. The overall time complexity of Algorithm 1 is $O(m \times n^2)$. Besides, the Apriori algorithm directly provides all probability values for the construction of EDG.

3.2 Calculation of Weights

Given the structure of graph G_E, it is necessary to accurately quantify the weights of edges by further exploring the co-occurrences statistics from data. Thus, we introduce *coefficient of association* $coa(v_i, v_j)$ as the weight $w_{ij} \in W$ of each edge $\overline{v_i v_j}$. According to the intuition of $coa(v_i, v_j)$, the following properties should be satisfied:

- *Symmetry*: $coa(v_i, v_j) = coa(v_j, v_i)$.
- *Non-negativity*: $coa(v_i, v_j) > 0$.
- *Identical boundedness*: $coa(v_i, v_j) \leq 1$, $coa(v_i, v_j) = 1$ only if $v_i = v_j$.
- *Informativeness of co-occurrence*: The fewer occurrences of ψ in $\Psi(\vec{q})$ containing an entity pair (v_i, v_j), the more informative the (v_i, v_j) is, corresponding to a higher $coa(v_i, v_j)$.

- *Specificness of entity frequency*: Entity frequency (EF) denotes the proportion of an entity to the total number of entities in $\Psi(\vec{q})$. The greater the difference in frequency between v_i and v_j, the smaller the $coa(v_i, v_j)$.

To compute $coa(v_i, v_j)$, we first consider the *informativeness* of co-occurrence by describing the ratio of the number of co-occurrence entries for entity pairs in $\Psi(\vec{q})$:

$$\ln \frac{SN\left(\Psi(\vec{q})\right)}{TN\left(v_i, v_j\right)} \tag{3}$$

where $SN\left(\Psi(\vec{q})\right)$ denotes the total number of *transactions* in $\Psi(\vec{q})$, and $TN\left(v_i, v_j\right)$ represents the number of entity sequences containing both v_i and v_j.

Actually, we aim to distinguish the importance of different entity pairs by *informativeness* of co-occurrence. If (v_i, v_j) appears frequently and dispersedly in multiple entity sequences, we consider the co-occurrence of (v_i, v_j) is trivial and less informative. In contrast, if v_i and v_j co-occur in a smaller number of entity sequences, the associations between them are more representative and informative, which leads to a larger strength. Next, we consider the difference of frequency at the single entity level:

$$\exp\left|EF\left(v_i\right) - EF\left(v_j\right)\right| \tag{4}$$

where $EF\left(v_i\right)$ and $EF\left(v_j\right)$ means entity frequency of v_i and v_j respectively.

Equation (4) takes the *specificness* of entity frequency into account. The smaller the difference between $EF(v_i)$ and $EF(v_j)$ the closer of v_i and v_j. We choose exponential function to ensure that the overall value of Eq. (4) is a number greater than or equal to 1. At the same time, the trend of Eq. (4) is positively correlated with the frequency difference between v_i and v_j.

To combine Eq. (3) and Eq. (4) to jointly measure the weights of edges, we form Eq. (5) to reasonably reflect both trends. The unnormalized weight of $\overline{v_i v_j}$ is defined as follows:

$$\xi(v_i, v_j) = \frac{\ln \frac{SN(\Psi(\vec{q}))}{TN(v_i, v_j)}}{\exp\left|EF\left(v_i\right) - EF\left(v_j\right)\right|} \tag{5}$$

Here, the upper bound of $\xi(v_i, v_j)$ is not constrained, which does not facilitate our specific comparison between the weights of any two edges. The sum of all $\xi(v_i, v_j)$ in G_E is specified as follows:

$$Sum(G_E) = \sum_{v_i, v_j \in V, i \neq j} \xi(v_i, v_j) \tag{6}$$

Then, $\xi(v_i, v_j)$ could be normalized by combining Eq. (5) and Eq. (6).

$$coa(v_i, v_j) = \frac{\xi(v_i, v_j)}{Sum(G_E)} \tag{7}$$

We measure the $coa(v_i, v_j)$ individually to get the actual weights w_{ij} of each edge $\overline{v_i v_j} \in E$. Finally, the set of weights W of EDG could be obtained.

3.3 Ranking Associative Entities

To measure the association strength of any two entities in the EDG, we transform the nodes of G_E into low-dimensional vector space by graph embedding.

Specifically, given an EDG $G_E = (V, E, W)$, we learn the co-occurrence features and the neighboring relations among nodes in two steps: random walk sampling and skip-gram.

We use a tunable bias random walk mechanism [10] in the procedure of neighborhood sequences sampling. Let $v_s \in V$ be a source node, and c_l be the lth node in the walk, $c_0 = v_s$. The unnormalized transition probability $\pi(c_l, c_{l+1})$ is:

$$\pi(c_l, c_{l+1}) = \eta_{mn}(c_{l-1}, c_{l+1}) \times coa(c_l, c_{l+1}) \tag{8}$$

where $\eta_{mn}(c_{l-1}, c_{l+1})$ is a hyper parameter determined by the shortest path distance between c_{l-1} and c_{l+1}. m and n are user-defined parameters to control the bias of random walk. Then, the actual transition probability from v_i to v_j is κ_t, defined as follows:

$$\kappa_t = (c_l = v_j | c_{l-1} = v_i) = \begin{cases} \frac{\pi(v_i, v_j)}{Z} & \overline{v_i v_j} \in E \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where Z is the normalizing constant.

Upon the sample sequences, we aim to map each $v_i \in V$ into the same space: $f : v_i \longrightarrow \mathbb{R}^d$ (equivalent to Eq. (1)) by maximizing the log-probability function:

$$\max_f \sum_{v_i \in V} log[p(N_b(v_i)|f(v_i))] \tag{10}$$

where $N_b(v_i) \subset V$ is the network neighboring [10] of v_i generated by the random walk sampling strategy controlling by Eq. (8).

A matrix $H^{|V| \times d}$ could be obtained by Eq. (10). Each entry of $H^{|V| \times d}$ represents the vector of a specific entity in EDG. The association strength $ad(v_i, \vec{q})$ of v_i to $\vec{q}$ in EDG could be measured by the cosine similarity of vectors in $H^{|V| \times d}$:

$$ad(v_i, \vec{q}) = \frac{\sum_{j=1}^{d} H_{ij} \times H_{\vec{q}}}{\sqrt{\sum_{j=1}^{d} H_{ij}^2 \times \sum_{j=1}^{d} H_{\vec{q}}^2}} \tag{11}$$

where $H_{\vec{q}}$ denotes the vector representation of query $\vec{q}$. Note that if there are more than one entities in $\vec{q}$, the final $ad(v_i, \vec{q})$ is the average of similarities between the vector of v_i to the vectors of different entities in $\vec{q}$.

Finally, we could obtain a top-k ranked list $L(\vec{q}) = \{ad(v_1, \vec{q}), ..., ad(v_k, \vec{q})\}$, where $ad(v_i, \vec{q})(1 \leq i \leq n)$ is the ith maximal value in $L(\vec{q})$.

4 Experiments

In this section, we present experimental results on two real-life datasets to evaluate our proposed method. We first introduce the experimental settings, and then

conduct three sets of experiments: (1) ranking associative entities, (2) entity relatedness ranking, and (3) impacts of parameters to evaluate our method compared with existing methods.

4.1 Experiment Settings

Datasets. We perform experiments on two widely used datasets for evaluating entity relatedness, KORE [13] and ERT [12], and extract the datasets containing associative Wikipedia articles and Web pages from search engines.

Table 2. Statistics of datasets.

Dataset	Query entities	Candidate entities	Wikipedia articles	Google & Bing URLs
KORE	21	420	4,200	12,600
ERT	40	937	8,000	24,000

- **KORE**, extracts entities from YAGO2 covering four popular domains: IT companies, Hollywood celebrities, video games, and television series. For each query entity, 20 candidate entities linked to the query's Wikipedia article are ranked in descending order of human rating association scores and regarded as the ground-truth of the most relevant entities to the query.
- **ERT**, consists of query entity pairs within two topics: the first 20 groups are from the music Website last.fm, and the last 20 groups originate in the movie dataset IMDb. Several to dozens of candidate entities with association scores are given for each entity pair. The scores are computed by considering multiple properties of entities from DBpedia

To generate $D_{\vec{q}}$ of each query, we extract the associative texts of all query entities in the two datasets from search engines and Wikipedia. Specifically, we first crawl the Web pages of the top 300 URLs from Google and Bing by using the query entities as queries. We then collect the top-ranked 200 Wikipedia articles by inputting the query entities into the Wikipedia dump. We finally combine these text contents as $D_{\vec{q}}$. Note that we also pre-process these texts by removing redundancy and building indices. The important statistics about these datasets are summarized in Table 2.

Evaluation Metrics. Three groups of metrics are chosen in our experiments: (1) Normalized discounted cumulative gain (NDCG) [20] to evaluate the accuracy of each entity ranking method, (2) Pearson, Spearman correlation coefficients and their harmonic mean [25] to evaluate the consistency between ranking results and ground truth, and (3) precision, recall and F1-score to evaluate the effectiveness of EDG with varying the user-defined threshold of minimal-support σ.

Comparison Methods. We choose six methods for comparison with our EDGM.

- Milne-Witten (MW) [27] is a typical graph-based approach to measure associations between entities using hyperlink structures of Wikipedia.
- ESA [8] is a representative text-based method by using entity co-occurrence information and TF-IDF weights.
- Entity2vec (E2V) [28] jointly learns vector representations of words and entities from Wikipedia by the skip-gram model.
- TSF [25] is a two-stage entity relevance computing framework for Wikipedia by first generating a weighted subgraph for co-occurrence information and then computing the relatedness on the subgraph.
- E-PR is the EDG with PageRank [22] to rank the associative entities.
- E-DW is the EDG with the DeepWalk for graph embedding.

Implementation. To generate *transactions* in our EDGM, we use the tool WAT [24] to link entities in $D_{\bar{q}}$ to their corresponding entity-IDs in Wikipedia, and filter the Top-300 matching candidates with the highest similarity to the given query based on KG embedding by OpenKE.[1] For ESA and MW, we take the current query and candidate entities in the corresponding EDG as input, and generate the ranking list based on the relatedness between the candidate entities and the query. For E2V, we obtain the representations of words and entities based on the same version of Wikipedia chosen for EDGM. For TSF, we adopt the recommended configurations [25] to achieve the optimal results. We transform the undirected edges of EDG to bidirectional directed edges for E-PR and perform E-DW on the unweighted graph structure of EDG[2].

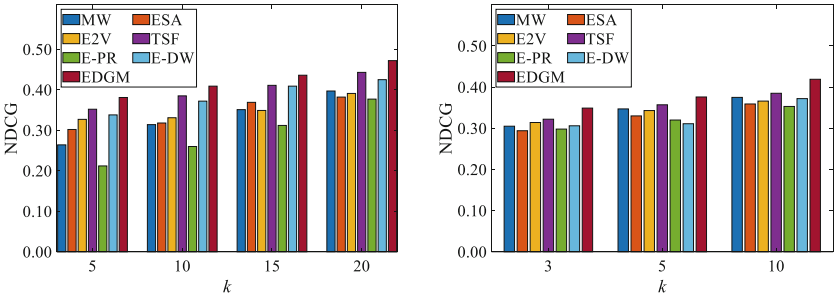
To balance the effectiveness and efficiency of EDG construction, we fix the threshold σ to 11 on KORE and 6 on ERT. For EDGs constructed by each query, they contain an average of 37 entity nodes on KORE and 49 entity nodes on ERT. We also set the node2vec parameters $dimensions, walklength, numberofwalks$ to 128, 30 and 200 on KORE and 128, 30 and 100 on ERT for better graph embedding. Besides, we find that the BFS random walk strategy ($m = 1$, $n = 2$) is more conducive to achieving the best results for our model.

4.2 Experimental Results

Exp-1: Ranking Associative Entities. To test the accuracy of associative entities ranking by our EDGM, we record NDCG of the top-k ranked lists found by all methods when k is fixed to 5, 10, 15 and 20 on KORE and 3, 5 and 10 on ERT. The results are shown in Fig. 3(a) and Fig. 3(b) respectively. All methods rank the entities that exist in the current EDG, and missing entities are ignored and skipped.

The results tell us that (1) our method EDGM achieves the highest NDCG scores and outperforms other methods on all datasets by taking the advantages of weighted associations between entity nodes in EDG, (2) our EDGM performs consistently better than other methods on all datasets by presenting the frequency characteristics including *informativeness* of co-occurrence and *specificness* of entity frequency, while some methods perform unstably on different

[1] http://139.129.163.161/index/toolkits.
[2] https://github.com/opp8888/ConstructionofEDG.



(a) NDCG on KORE (b) NDCG on ERT

Fig. 3. Results of associative entities ranking.

datasets. For example, ESA performs better than MW in KORE but works worse than MW in ERT, and (3) E-DW is better than E-PR, which indicates that graph embedding is effective for our entity ranking model, and EDGM outperforms E-DW, which also suggests that node2vec is more suitable for embedding the EDG than DeepWalk. In fact, our EDGM improves NDCG by 6.7% and 7.5% over the second-highest method TSF on KORE and ERT, respectively. This verifies the effectiveness of our proposed method.

Table 3. Comparison of entity relatedness ranking on KORE.

Domain	Metrics	MW	ESA	E2V	TSF	E-PR	E-DW	EDGM
IT companies	Pearson	0.496	0.489	0.579	**0.753**	0.192	0.652	0.749
	Spearman	0.537	0.664	0.613	0.741	0.425	0.688	**0.767**
	Harmonic	0.516	0.563	0.596	0.747	0.265	0.670	**0.758**
Hollywood celebrities	Pearson	0.515	0.577	0.675	0.727	0.216	0.613	**0.811**
	Spearman	0.634	0.692	0.589	0.792	0.372	0.582	**0.805**
	Harmonic	0.568	0.629	0.629	0.758	0.273	0.597	**0.808**
Video games	Pearson	0.607	0.552	0.616	0.781	0.18	0.587	**0.793**
	Spearman	0.592	0.621	0.542	**0.810**	0.489	0.675	0.791
	Harmonic	0.599	0.584	0.577	**0.795**	0.263	0.628	0.792
Television series	Pearson	0.671	0.521	0.637	**0.833**	0.261	0.712	0.691
	Spearman	0.735	0.585	0.671	0.732	0.491	0.716	**0.754**
	Harmonic	0.702	0.551	0.654	**0.779**	0.341	0.714	0.721

Exp-2: Entity Relatedness Ranking. Exp-2 aims to test whether our EDGM could generate the ranking lists having a high degree of consistency compared with the ground truth. The results on KORE and ERT are shown in Table 3 and Table 4, respectively. Since the number of entities of EDG are not fixed, the top-5 candidate entities in the current EDG are selected for discussion.

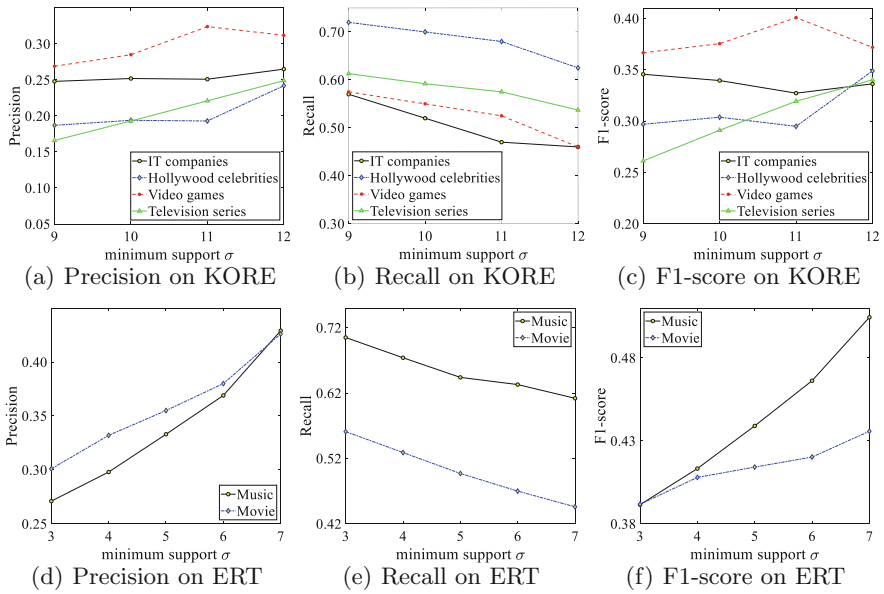
The results tell us that (1) EDGM performs better than the traditional text-based, graph-based methods (MW and ESA) and the pure entity representation approach (E2V) in all domains of the two datasets, (2) EDGM outperforms TSF in most domains of ERT and performs as well as TSF on KORE, and (3) EDGM achieves the highest harmonic mean in most domains of the two datasets. Our

Table 4. Comparison of entity relatedness ranking on ERT.

Domain	Metrics	MW	ESA	E2V	TSF	E-PR	E-DW	EDGM
Music	Pearson	0.677	0.531	0.652	**0.795**	0.257	0.694	0.781
	Spearman	0.589	0.663	0.598	0.732	0.386	0.660	**0.787**
	Harmonic	0.630	0.590	0.624	0.762	0.309	0.677	**0.784**
Movie	Pearson	0.615	0.466	0.681	**0.828**	0.190	0.785	0.825
	Spearman	0.463	0.569	0.626	0.764	0.429	0.682	**0.771**
	Harmonic	0.528	0.512	0.652	0.795	0.263	0.730	**0.797**

EDGM gives better results in total rank, which verifies the effectiveness of EDG that generates a powerful presentation of the associations upon neighboring and co-occurrence features of entities.

Exp-3: Impacts of Parameters. To evaluate the impacts of the threshold σ, we vary σ from 9 to 12 on KORE and from 3 to 7 on ERT. The results are reported in Fig. 4(a)–Fig. 4(f), respectively.



(a) Precision on KORE (b) Recall on KORE (c) F1-score on KORE

(d) Precision on ERT (e) Recall on ERT (f) F1-score on ERT

Fig. 4. Results of impacts of parameters.

The results tell us that (1) the precision increases (recall decreases) with the increase of σ, which is consistent with the theoretical expectation that the number of entity nodes in EDG decrease when σ increases, and (2) the F1-score remains relatively stable when varying σ, which demonstrates that our method

could efficiently recall candidate entities in the ground truth. Note that our model achieves better recall than precision, which is suitable for the application scenarios of ranking problems requiring a higher recall. Hence, we fix σ to 11 on KORE and 6 on ERT to balance the size of EDG and guarantee high recall.

5 Conclusions and Future Work

In this paper, we propose the entity dependency graph model (EDGM) to rank associative entities in KG by graph embedding upon frequent entities. By incorporating multiple features of the association from both KG and Web resources effectively, one or more entities are allowed as a query to achieve better scalability. EDGM facilitates the discovery of associative entities with high recall, since the co-occurrence of entities in KG and the behavioral associations could be represented by a global model in an unsupervised manner.

However, the path and label information in KG, as well as the impacts of neighbors in a random walk on EDGM have not been well considered, which needs further exploration. Moreover, open world KG completion is worthwhile to study further by incorporating with semantic/implicit associations between entities achieved by our method.

Acknowledgements. This paper is supported by National Natural Science Foundation of China (U1802271, 62002311), Science Foundation for Distinguished Young Scholars of Yunnan Province (2019FJ011), China Postdoctoral Science Foundation (2020M673310), Program of Donglu Scholars of Yunnan University. We thank Prof. Weiyi Liu from Yunnan University for his insightful advice.

References

1. Aggarwal, C.C., Bhuiyan, M.A., Hasan, M.A.: Frequent pattern mining algorithms: a survey. In: Aggarwal, C.C., Han, J. (eds.) Frequent Pattern Mining, pp. 19–64. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07821-2_2
2. Aggarwal, N., Buitelaar, P.: Wikipedia-based distributional semantics for entity relatedness. In: AAAI (2014)
3. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans. Knowl. Data Eng. **30**(9), 1616–1637 (2018)
4. Chen, J., Gu, B., Li, Z., Zhao, P., Liu, A., Zhao, L.: SAEA: self-attentive heterogeneous sequence learning model for entity alignment. In: DASFAA, pp. 452–467 (2020)
5. Cheng, G.: Relationship search over knowledge graphs. SIGWEB Newsl., 8 p. (2020). https://doi.org/10.1145/3409481.3409484. Article ID 3
6. Cheng, G., Liu, D., Qu, Y.: Fast algorithms for semantic association search and pattern mining. IEEE Trans. Knowl. Data Eng. **33**(04), 1490–1502 (2019). https://doi.org/10.1109/TKDE.2019.2942031. ISSN 1558–2191
7. Cheng, G., Shao, F., Qu, Y.: An empirical evaluation of techniques for ranking semantic associations. IEEE Trans. Knowl. Data Eng. **29**(11), 2388–2401 (2017)

8. Gabrilovich, E., Markovitch, S.: Wikipedia-based semantic interpretation for natural language processing. J. Artif. Intell. Res. **34**, 443–498 (2009)
9. Gong, X., Xu, H., Huang, L.: Han: Hierarchical association network for computing semantic relatedness. In: AAAI (2018)
10. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD, pp. 855–864 (2016)
11. Han, J., Kamber, M., Pei, J.: Data Mining (Third Edition). Morgan Kaufmann, Burlington (2012)
12. Herrera, J.E.T., Casanova, M.A., Nunes, B.P., Leme, L.A.P.P., Lopes, G.R.: An entity relatedness test dataset. In: ISWC, pp. 193–201 (2017)
13. Hoffart, J., Seufert, S., Nguyen, D.B., Theobald, M., Weikum, G.: KORE: keyphrase overlap relatedness for entity disambiguation. In: CIKM, pp. 545–554 (2012)
14. Imrattanatrai, W., Kato, M.P., Tanaka, K., Yoshikawa, M.: Entity ranking for queries with modifiers based on knowledge bases and web search results. IEICE Trans. Inf. Syst. **101**(9), 2279–2290 (2018)
15. Li, J., Chen, W., Gu, B., Fang, J., Li, Z., Zhao, L.: Measuring semantic relatedness with knowledge association network. In: DASFAA, pp. 676–691 (2019)
16. Li, L., Yue, K., Zhang, B., Sun, Z.: A probabilistic approach for inferring latent entity associations in textual web contents. In: DASFAA, pp. 3–18 (2019)
17. Liu, W., Yue, K., Wu, H., Fu, X., Zhang, Z., Huang, W.: Markov-network based latent link analysis for community detection in social behavioral interactions. Appl. Intell. **48**(8), 2081–2096 (2017). https://doi.org/10.1007/s10489-017-1040-y
18. Liu, Z., et al.: Distance-aware dag embedding for proximity search on heterogeneous graphs. In: AAAI, pp. 2355–2362 (2018)
19. Navigli, R., Martelli, F.: An overview of word and sense similarity. Nat. Lang. Eng. **25**(6), 693–714 (2019)
20. Nguyen, T., Tran, T., Nejdl, W.: A trio neural model for dynamic entity relatedness ranking. In: CoNLL, pp. 31–41 (2018)
21. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. Commun. ACM **62**(8), 36–43 (2019)
22. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab (1999)
23. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: SIGKDD, pp. 701–710 (2014)
24. Piccinno, F., Ferragina, P.: From TagME to WAT: a new entity annotator. In: The First International Workshop on Entity Recognition and Disambiguation, pp. 55–62 (2014)
25. Ponza, M., Ferragina, P., Chakrabarti, S.: On computing entity relatedness in wikipedia, with applications. Knowl.-Based Syst. **188**, 105051 (2020)
26. Schuhmacher, M., Dietz, L., Ponzetto, S.P.: Ranking entities for web queries through text and knowledge. In: CIKM, pp. 1461–1470 (2015)
27. Witten, I.H., Milne, D.N.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: AAAI (2008)
28. Yamada, I., Shindo, H., Takeda, H., Takefuji, Y.: Joint learning of the embedding of words and entities for named entity disambiguation. In: SIGNLL, pp. 250–259 (2016)
29. Zhang, Y., Wang, D., Zhang, Y.: Neural IR meets graph embedding: a ranking model for product search. In: WWW, pp. 2390–2400 (2019)