

# NERank: Bringing Order to Named Entities from Texts

Chengyu Wang<sup>1</sup>, Rong Zhang<sup>1</sup>, Xiaofeng He<sup>1(✉)</sup>, Guomin Zhou<sup>2</sup>,  
and Aoying Zhou<sup>1</sup>

<sup>1</sup> Institute for Data Science and Engineering,  
East China Normal University, Shanghai, China  
chywang2013@gmail.com, {rzhang,xfhe,ayzhou}@sei.ecnu.edu.cn  
<sup>2</sup> Zhejiang Police College, Hangzhou, Zhejiang Province, China  
zhouguomin@zjjcxy.cn

**Abstract.** Most entity ranking research aims to retrieve a ranked list of entities from a Web corpus given a query. However, entities in plain documents can be ranked directly based on their relative importance, in order to support entity-oriented Web applications. In this paper, we introduce an entity ranking algorithm NERank to address this issue. NERank first constructs a graph model called Topical Tripartite Graph from a document collection. A ranking function is designed to compute the prior ranks of topics based on three quality metrics. We further propose a meta-path constrained random walk method to propagate prior topic ranks to entities. We evaluate NERank over real-life datasets and compare it with baselines. Experimental results illustrate the effectiveness of our approach.

**Keywords:** Entity ranking · Topic modeling · Topical tripartite graph · Meta-path constrained random walk

## 1 Introduction

Ranking problems have been extensively studied to bring order to varying types of objects, such as Web pages [1], products [2] and textual units [3]. With the number of entities increasing rapidly on the Web, the problem of Entity Ranking (ER) has drawn much attention. For example, ER tracks have been conducted in INEX and TREC since 2006 and 2009, to rank entities from Web corpora given a query topic [4].

In traditional ER tasks, the rank of entities is measured by the relevance between a query topic (e.g. *impressionist art in the Netherlands* [5]) and entities

---

A preliminary version of this paper has been presented in WWW'16 [6]. This work is partially supported by NSFC under Grant No. 61402180, the Natural Science Foundation of Shanghai under Grant No. 14ZR1412600, Shanghai Agriculture Science Program (2015) Number 3-2 and NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant No. U1509219.

with contextual information. In this paper, we consider a different problem: *ranking entities in document collections based on the importance of entities*. For example, given news articles related to *Egypt Revolution* as input, ER aims to retrieve a ranked list of entities that are most relevant to *Egypt Revolution*, including people (e.g., *Hosni Mubarak*, *Mohamed Morsi*), locations (e.g., *Egypt*, *Cairo*), organizations (e.g., *Muslim Brotherhood*), etc<sup>1</sup>.

The task of ER in this paper is vital for several Web-scale applications:

- **Entity-oriented Web Search:** It facilitates Web entity recommendation, rather than retrieving a list of Web documents that are relevant to the user query but contain abundant or irrelevant information.
- **Web Semantification:** It identifies important entities from Web documents and add semantic tags to the Web automatically.
- **Knowledge Base Population:** It potentially improves the performance of knowledge base population by extracting and ranking entities from the Web and linking them to knowledge bases.

The challenge of ER is that the ranking order of entities should be determined by the contents of the document collection, with no additional knowledge sources or user queries available. Additionally, the importance of entities is expressed implicitly in natural language text, which can not be measured directly. Therefore, it is difficult to extend traditional ER techniques to this scenario.

In this paper, we introduce a graph-based ranking algorithm *NERank* to solve this task. Given a document collection as input, we mine latent topics and model the semantic relations between documents, topics and entities in a graphical model called *Topical Tripartite Graph* (TTG). A ranking function is designed to estimate prior ranks of topics via three quality metrics (i.e., prior probability, entity richness and topic specificity). The prior ranks are propagated along paths in the TTG via a meta-path constrained random walk algorithm. The final rank of entities can be estimated when this process converges.

In summary, we make the following contributions in this paper:

- We formalize the ER problem. A graphical structure TTG is proposed to model the implicit semantic relations between documents, topics and entities.
- A ranking function is designed to calculate the prior ranks of topics based on three quality metrics. We introduce a meta-path constrained random walk algorithm to compute the ranks of entities by rank propagation.
- We conduct extensive experiments and case studies to illustrate the effectiveness of our approach.

The rest of this paper is organized as follows. Section 2 summarizes the related work. We define the ER problem in Sect. 3. The proposed approach is described in Sects. 4 and 5. Experimental results are presented in Sect. 6. We conclude our paper and discuss the future work in Sect. 7.

---

<sup>1</sup> See background info at:

[https://en.wikipedia.org/wiki/Egyptian\\_Revolution\\_of\\_2011](https://en.wikipedia.org/wiki/Egyptian_Revolution_of_2011).

## 2 Related Work

Research efforts on ER have been put to address the problem of retrieving a ranked list of entities given a query. In the task of ER, entities can be of a certain type, for example, searching for experts in a specific domain [7]. The more general problem is ranking entities of various kinds. Recently, a lot of ER related research has been conducted in the context of INEX and TREC evaluation, started in 2006 and 2009, respectively.

Besides these ER tracks, ER provides a new paradigm to rank and retrieve information at an entity level in the field of Web search. Nie et al. [8] propose a link analysis model PopRank to rank Web “objects” (i.e., entities) within a specific domain, which considers the relevance and popularity of entities. For vertical search, Ganesan et al. [2] leverage online reviews to design several ER models based on user’s preference for the purpose of product ranking and recommendation. Lee et al. [9] model multidimensional recommendation as an ER problem, and adopt Personalized PageRank algorithm [10] to rank entities for e-commerce applications.

External data sources are utilized to provide additional information for more accurate ER. Kaptein et al. [4] use the Wikipedia category structure as a pivot to identify key entities properly. They reduce the problem of Web ER to Wikipedia ER. Ilieva et al. [11] make use the rich attribute information in knowledge bases to improve the coverage and quality of ER. For short text analysis, Meij et al. [12] apply learning-to-rank models to extract key concepts in tweets and link to Wikipedia. However, none of the prior work considers the ER task in this paper. With entities in documents ranked correctly, a series of Web applications can be benefitted to provide entity-oriented service.

## 3 Entity Ranking Problem

According to the task setting of ER, we take a collection of documents (denoted as  $D$ ) as input. Let  $m \in M$  denote an entity mention in document  $d \in D$ , recognized by Named Entity Recognition (NER) techniques. Because entity mentions appeared in the plain texts are unnormalized, simply ranking on  $M$  will result in the “unnormalized ranking” issue. Consider the example in Table 1. Both

**Table 1.** Comparison between unnormalized and normalized ranking.

Unnormalized ranking		Normalized ranking	
Entity mention	Rank value	Normalized entity	Rank value
Egypt	0.25	Hosni Mubarak	0.35
Mubarak	0.2	Egypt	0.25
Hosni Mubarak	0.15	Cario	0.1
Cario	0.1	...	...

“Hosni Mubarak” and “Mubarak” refer to the former Egypt president “Hosni Mubarak”. If they are unnormalized, they will receive separate, inconsistent and under-estimated rank values.

Therefore, we employ an entity normalization procedure to map each  $m \in M$  to its normalized form  $e \in E$ . To accomplish the task of ER, we assign each entity  $e \in E$  a rank  $r(e)$  to represent the relative importance in  $D$ . For the illustration purpose, the high-level process of ER is presented in Fig. 1. We also provide a simple example to show the data processing steps of ER. Here, we present the definition of ER formally as follows:

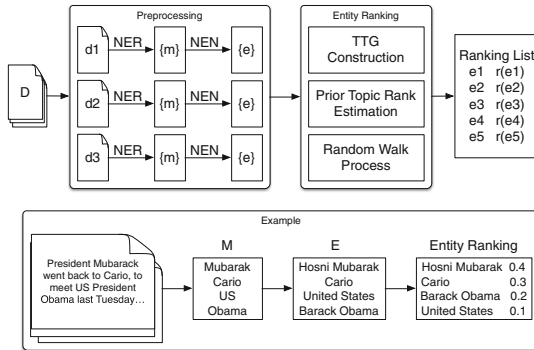


Fig. 1. Illustration of the ER process and a simple example.

**Definition 1. Entity Ranking.** Given a document collection  $D$  and a normalized named entity collection  $E$  detected from  $D$ , the goal is to give each entity  $e \in E$  a rank  $r(e)$  to denote the relative importance such that (1)  $0 \leq r(e) \leq 1$  and (2)  $\sum_{e \in E} r(e) = 1$ .

The task definition in this paper is similar to the task *Ranked-concepts to Wikipedia* (Rc2W) [13] and the more general task *Ranked-concepts to Knowledge Base* (Rc2KB) in the entity annotator benchmark GERBIL [14]. We notice that both tasks are comprised of two sub-steps: (i) entity linking (which maps an entity mention to an entity in knowledge base) and (ii) ranking (which generates the ranked order of entities). While much of the previous work addressed the task of entity linking, we focus more on ER, which is not sufficiently studied. Another difference is that since existing knowledge bases still face the incompleteness issue, we do not require entities to be linked to Wikipedia or a knowledge base before they can be ranked.

## 4 Topical Tripartite Graph Modeling

The key for accurate ER is to mine the implicit semantic relations between documents and entities. Extracting language patterns that can help identify

important entities from texts is difficult, due to the flexibility and complexity in expression of natural languages. However, by topic modeling, the gap between documents and entities can be bridged. In this section, we introduce the formal definition of TTG and show the construction process of the graph in detail.

*Topical Tripartite Graph.* The TTG is a tripartite graph to model the semantic relations among  $\langle \text{document}, \text{topic} \rangle$  and  $\langle \text{topic}, \text{entity} \rangle$  pairs. There are three types of nodes (i.e., documents  $D$ , topics  $T$  and (normalized) entities  $E$ ), and two types of weighted, undirected edges (i.e., document-topic edges  $R_{DT}$  and topic-entity edges  $R_{TE}$ ). Here, we give the formal definition of TTG as follows:

**Definition 2. Topical Tripartite Graph.** A TTG w.r.t. document collection  $D$  is a weighted, tripartite graph  $G_D = (D, T, E, R_{DT}, R_{TE})$ . The nodes of the graph are partitioned into three disjoint sets: documents  $D$ , topics  $T$  and entities  $E$ .  $R_{DT}$  and  $R_{TE}$  are edge sets that connect nodes in  $\langle D, T \rangle$  and  $\langle T, E \rangle$  pairs, respectively.

Additionally, weights of edges in TTG can be employed to quantify the degrees of relation strength. In this paper, we employ a weight  $w_{dt}(d_i, t_j) \in (0, 1)$  for an edge  $(d_i, t_j) \in R_{DT}$  and  $w_{te}(t_i, e_j) \in (0, 1)$  for an edge  $(t_i, e_j) \in R_{TE}$ .

*Graph Construction.* The TTG construction process includes two parts: (1) *named entity recognition and normalization*, which discovers and normalizes named entities in document collections, and (2) *entity-aware topic modeling*, which mines the latent topics in documents and calculates the weights of edges in TTG. The general construction process of TTG is shown in Fig. 2.

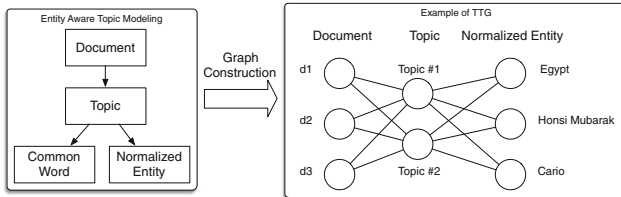


Fig. 2. Illustration of the TTG construction process.

**Named Entity Recognition and Normalization.** Entities in documents can be automatically recognized by the NER tagger, such as Conditional Random Fields [15]. Before we construct the TTG, entity normalization is necessary to transform entity mentions recognized by NER to normalized forms. Take the sentence “Mubarak was an former political leader in Egypt” as an example. It is processed by an NER tagger, shown as follows:

*Mubarak*#PERSON was a political leader in *Egypt*#LOCATION

After entity normalization, the tagging sequence becomes:

*Hosni Mubarak*#PERSON was a political leader in *Egypt*#LOCATION

In this paper, we employ the algorithm proposed by Jijkoun et al. [16] for entity normalization, which applies approximate name matching, identification of missing references and name disambiguation techniques. Due to space limitations, we omit the details here.

**Entity Aware Topic Modeling.** Topic models such as LDA [17] can model the latent topics in documents. However, LDA models a document using the “bag-of-words” model, without taking multi-word entities or unnormalized entity mentions into consideration. To better fit the ER task, we introduce an *entity aware topic modeling* approach, which models documents as a collection of textual units, consisting of *normalized entities* and *common words* (denoted as  $W$ ). Additionally, we remove stop words and punctuations in these documents. Following the previous example, given sentence “Mubarak was a political leader in Egypt”, we treat “Hosni Mubarak” and “Egypt” as normalized entities, and “political” and “leader” as common words.

With entity normalization and preprocessing steps, LDA is employed to model the document-topic distributions  $\Theta$  (represented as a  $|D| \times |T|$  matrix) and the topic-textual unit distributions  $\Phi$  ( $|T| \times |E \cup W|$  matrix) given the document collection  $D$ . In Table 2, we present some topics we discovered in the document collection w.r.t *Egypt Revolution*. We also manually add a description of each topic to illustrate that this approach is effective to detect latent aspects in the document collection and model the relations between topics and entities.

The weights of edges in TTG are assigned based on distributions of entity aware topic modeling. If the probability of a topic is high in a document, it means the topic and the document have strong semantic associativity. Therefore, for document  $d_i$  and topic  $t_j$ , the weight is defined as  $w_{dt}(d_i, t_j) = \theta_{i,j}$  where  $\theta_{i,j}$  is the element in the  $i^{th}$  row and the  $j^{th}$  column of  $\Theta$ . Similarly, the semantic relations between topics and entities can be measured by the topic-textual unit distribution. We remove the columns for topic-common word distributions in  $\Phi$ , and denote the rest part of the matrix as  $\hat{\Phi}$  (called topic-entity matrix). For topic  $t_i$  and entity  $e_j$ , we have  $w_{te}(t_i, e_j) = \hat{\phi}_{i,j}$  where  $\hat{\phi}_{i,j}$  is the element in the  $i^{th}$  row and the  $j^{th}$  column of  $\hat{\Phi}$ .

**Table 2.** Topics discovered in document collection w.r.t. *Egypt Revolution*.

Topic	Top normalized entities	Top common words	Description
#1	Egypt, Hosni Mubarak	political, military, revolution	Start of the revolution
#2	Mohamed Morsi, Egypt	President, constitution, vote	Presidential election
#3	Egypt, Israel, Iran	government, foreign, peace	Foreign countries’ reaction
#4	Egypt, Cairo	economic, government, billion	Revolution’s effect on economy
#5	Egypt	tourism, tourist, travel, sea	Revolution’s effect on tourism

## 5 Entity Ranking Algorithm

In this section, we present our *NERank* algorithm. Based on TTG, we compute the prior ranks of topics in combination of three quality metrics. After that, a

meta-path constrained random walk algorithm is proposed to calculate the ranks of entities by propagating the prior topic ranks to entities over the TTG.

### 5.1 Prior Topic Rank Estimation

Without additional knowledge sources, it is difficult to determine the relative importance of documents and entities. In contrast, entity aware topic modeling can provide prior knowledge about topics. For example, in Table 2, we can see that Topic #1 and Topic #2 are directly about major events in *Egypt Revolution* and Topics #3-#5 discuss different aspects related to *Egypt Revolution*, but are less relevant. To facilitate ER, we design the following quality metrics to calculate the prior ranks of topics.

**Prior Probability.** Different topics have different probabilities to be discussed in documents. Some topics are related to more documents in  $D$  (e.g. Topic #1 in Table 2), while others are only related to only a few articles (e.g. Topic #5). We define the prior probability  $pr(t_i)$  of each topic  $t_i \in T$  using document-topic distributions as  $pr(t_i) = \frac{1}{|D|} \sum_{j=1}^{|D|} \theta_{j,i}$ . Because  $\sum_{j=1}^{|D|} \sum_{t=1}^{|T|} \theta_{j,t} = |D|$ ,  $|D|$  is served as a normalization factor.

**Entity Richness.** Entity richness measures the “goodness” of a topic from an entity aspect. As entities play an important role in documents, the “richness” of entities is a useful signal to measure the quality of topics. Here, we compute the “richness” as the sum of all probabilities of entities given topic  $t_i$ , i.e.,  $\sum_{j=1}^{|E|} \hat{\phi}_{i,j}$ . Therefore, the entity richness score for topic  $t_i$  is defined as:  $er(t_i) = \frac{1}{Z_{er}} \sum_{j=1}^{|E|} \hat{\phi}_{i,j}$  where  $Z_{er} = \sum_{m=1}^{|T|} \sum_{n=1}^{|E|} \hat{\phi}_{m,n}$  is a normalization factor.

**Topic Specificity.** Topic specificity measures the quality of a topic in an information theoretic approach. Based on the analysis on entities and common words in each topic, we observe that some topics are specific about some events or latent aspects, while others only provide background information. We extract all probabilities of topic  $t_i$  in all  $d \in D$  as a  $|D|$ -dimensional vector  $\langle \theta_{1,i}, \theta_{2,i}, \dots, \theta_{|D|,i} \rangle$ . The unnormalized “specificity” of topic  $t_i$  can be computed as  $ts'(t_i) = \sum_{j=1}^{|D|} \theta_{j,i} \log_2 \theta_{j,i}$ .

High “specificity” value means that there is no significant “burst” in topic distributions, which filters out topics that are only strongly related to few documents. However, if a topic rarely appears in any documents, it may receive a relatively high “specificity” score. In the implementation, we add a heuristic rule to avoid this problem: if the prior probability  $pr(t_i)$  is smaller than a small threshold  $\epsilon$ , we set  $ts(t_i) = 0$ . Hence, the topic specificity of  $t_i$  is defined as:

$$ts(t_i) = \begin{cases} 0 & pr(t_i) < \epsilon \\ \frac{1}{Z_{ts}} \sum_{j=1}^{|D|} \theta_{j,i} \log_2 \theta_{j,i} & pr(t_i) \geq \epsilon \end{cases}$$

where  $Z_{ts} = \sum_{i=1}^{|T|} ts(t_i)$  is a normalization factor.

**Ranking Function.** Combined the three quality metrics together, we can generate a feature vector for each topic  $t_i \in T$ , i.e.,  $\mathbf{F}(t_i) = \langle pr(t_i), er(t_i), ts(t_i) \rangle$ .

Denote  $\mathbf{W}$  as the weight vector where each element in  $\mathbf{W}$  gives different importance for different features such that  $\forall w_i > 0$  and  $\sum_i w_i = 1$ . Thus, the prior rank for topic  $t_i$  is defined as  $r_0(t_i) = \mathbf{W}^T \cdot \mathbf{F}(t_i)$ .

To learn the weights  $\mathbf{W}$  for the features, we employ the max-margin technique introduced in [18]. Given two topics  $t_i$  and  $t_j$ , if  $t_i$  is a more important topic than  $t_j$ , judged by human annotators, we have  $r_0(t_i) > r_0(t_j)$ . This implies that the following constraint holds:  $\mathbf{W}^T \cdot \mathbf{F}(t_i) - \mathbf{W}^T \cdot \mathbf{F}(t_j) \geq 1 - \xi_{i,j}$  where  $\xi_{i,j} \geq 0$  is a slack variable. This learning problem can be modeled as training a linear SVM classifier with the objective function  $\|\mathbf{W}\|_2^2 + C \cdot \sum_{i,j} \xi_{i,j}$ , where  $C$  is a tolerance parameter.

## 5.2 Meta-Path Constrained Random Walk Algorithm

With prior ranks of topics estimated, we aim to propagate ranks to other nodes in order to obtain entity ranks. Based on the graphical structure of TTG, we design a meta-path constrained random walk algorithm to rank entities.

In a TTG, we observe that (1) only topic nodes are connected with different types of nodes (i.e., documents and entities) and (2) we only have prior knowledge about ranks of topics. Thus, we define topic-centric meta-paths to constrain the behavior of random walkers. Denote  $x \rightarrow y$  as the action where the random surfer walks from  $x$  to  $y$ . We define two types of meta-paths to embed the semantics of document-topic and topic-entity relations, shown as follows:

**Definition 3. TDT Meta-path.** A TDT meta-path is a path defined over a TTG  $G_D$  which has the form  $t_i \rightarrow d_j \rightarrow t_k$  where  $t_i, t_k \in T$  and  $d_j \in D$ .

**Definition 4. TET Meta-path.** A TET meta-path is a path defined over a TTG  $G_D$  which has the form  $t_i \rightarrow e_j \rightarrow t_k$  where  $t_i, t_k \in T$  and  $e_j \in E$ .

TDT meta-paths encode the mutual enforcement effect between ranks of documents and topics. The assumption is that “good” documents relate to “good” topics and vice versa. TET meta-paths update the ranks of entities and pass the rank back to topic nodes for the next iteration of random walk.

Because random walk algorithms in meta-paths are effective for inference based on previous research [19], we compute the ranks of entities by meta-path constrained random walk. To better fit the graphical structure of a topical tripartite graph, we require that the random surfer is only allowed to walk along TDT and TET meth-paths. To specify, the random surfer begins by selecting a topic node  $t_i \in T$  with probability  $r_0(t_i)$  (i.e., the prior rank of  $t_i$ ) as the starting point. Next, the surfer makes the transfer along TDT and TET meta-paths. Denote  $\alpha$  and  $\beta$  as tuning parameters where  $\alpha > 0$ ,  $\beta > 0$  and  $\alpha + \beta < 1$ . One iteration of the random walk process is shown as follows:

- With probability  $\alpha$ , the random surfer walks through a TDT meta-path  $t_i \rightarrow d_j \rightarrow t_k$ .  $d_j$  is selected with probability  $\frac{\theta_{j,i}}{\sum_{d_k \in D} \theta_{k,i}}$  for all  $d_j \in D$ . Next,  $t_k$  is selected with probability  $\theta_{j,k}$  for all  $t_k \in T$ .



- With probability  $\beta$ , the random surfer walks through a TET meta-path  $t_i \rightarrow e_j \rightarrow t_k$ .  $e_j$  is selected with probability  $\frac{\hat{\phi}_{i,j}}{\sum_{e_k \in E} \hat{\phi}_{i,k}}$  for all  $e_j \in E$ . Next,  $t_k$  is selected with probability  $\frac{\hat{\phi}_{k,j}}{\sum_{t_m \in T} \hat{\phi}_{m,j}}$  for all  $t_k \in T$ .
- With probability  $1 - \alpha - \beta$ , the random surfer jumps to a topic node  $t_j$ .  $t_j$  is selected with probability  $r_0(t_j)$  for all  $t_j \in T$ .

This random walk process can be repeated iteratively until the system reaches equilibrium. Each entity node  $e_i$  will receive a score  $s(e_i)$ , indicating the number of visits by random surfers. Thus, the rank of an entity  $e_i$  is computed as  $r(e_i) = \frac{s(e_i)}{\sum_{e_j \in E} s(e_j)}$ . In the appendix, we will prove that this process will converge after a sufficient number of iterations, and give the close-form solution of *NERank*.

## 6 Experiments

In this section, we conduct extensive experiments on news datasets to evaluate the performance of *NERank*. We also compare our method with baselines to make the convincing conclusion.

### 6.1 Datasets and Experimental Settings

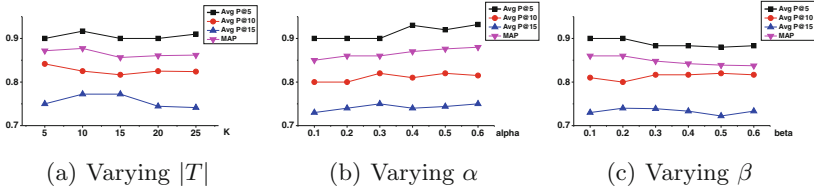
We use two publicly available news datasets in our experiments (i.e., **TimelineData** [20] and **CrisisData** [21]), described as follows:

- **TimelineData** - The dataset has 4,650 news articles that are related to 17 international events, such as *BP oil spill*, *death of Michael Jackson*, etc. Each group of news articles belongs to a news agency, such as BBC and CNN.
- **CrisisData** - The dataset contains 15,534 news articles that report four recent armed conflicts, including *Egypt Revolution*, *Syria War*, etc. These articles are from 24 news agencies, obtained using Google search engine.

To generate document collections, we randomly sample 100 documents from news articles related to the same event at each time. In total, we have 34 document collections from **TimelineData** and 16 from **CrisisData**. We conduct separate experiments on all document collections in the following experiments.

### 6.2 Experimental Results and Analysis

**Ground Truth.** The document collections used in this paper are all related to news events. For ground truth, we first obtain the news summaries of each document collection from [20,21], which are manually created by professional journalists. Based on the event summaries, we recruit a group of CS graduates to label the entities as “most important”, “important”, “relevant”, etc. Following the evaluation framework in [22], we finally have a ranked list of 15



**Fig. 3.** Evaluation results under different parameter settings.

entities w.r.t. a document collection by majority voting, which are regarded as “key” entities.

**Evaluation Metrics.** To evaluate different algorithms for ER, we compare the top- $k$  entities generated by machines with the ground truth. We employ Precision@ $K$  ( $K = 5, 10, 15$ ) and Average Precision as evaluation metrics. For multiple document collections, we take the average as results and report Average Precision@ $K$  and MAP in this paper. To compare *NERank* with baselines, we additionally use *paired t-test* to evaluate the level of statistical significance.

**Parameter Settings.** We tune parameters in *NERank*, namely, number of topics in LDA ( $|T|$ ) and parameters for random walk ( $\alpha$  and  $\beta$ ). In Fig. 3, we present the experimental results when we vary only one parameter at each time. In Fig. 3(a), we fix  $\alpha = \beta = 0.4$  and change the number of topics. It can be seen that although it is relatively hard to determine the number of topics, the performance of *NERank* is not sensitive to this issue. In Fig. 3(b) and (c), we set  $|T| = 10$  and one parameter ( $\alpha$  or  $\beta$ ) to be 0.4 and vary the other. It shows that our algorithm is not sensitive to the change of parameters. Note that the weight vector  $\mathbf{W}$  in the ranking function can be learned automatically and does not need to be tuned. We manually label 500 topic pairs to train the ranking model, and set  $|T| = 10$  and  $\alpha = \beta = 0.4$  in following experiments.

**Method Comparison.** To our knowledge, there is no prior work concerning ranking entities directly from document collections. However, there are abundant research on keyword extraction. In this paper, we take unsupervised keyword extraction methods as baselines. We first generate a ranked list of words using baselines and filter out common words in the list to produce the ranked list of entities. We also implement two variants of our approach, shown as follows:

- **TF-IDF** - rank entities based on TF-IDF scores.
- **TextRank** [3] - a graph-based iterative algorithm for textual unit ranking.
- **LexRank** [23] - a graph-based algorithm based on lexical centrality.
- **Kim et al.** [24] - a keyword extraction algorithm based on semantic similarity between words.
- **NERank<sub>Uni</sub>** - the variant of *NERank* which sets prior topic ranks uniformly.
- **NERank <sub>$\alpha=0$</sub>**  - the variant of *NERank* which sets  $\alpha = 0$  in random walk and thus ignores the semantic relatedness between documents and topics.

**Table 3.** Evaluation results for different methods. (\*: p-value $\leq$ 0.05)

Method	Average Precision@5	Average Precision@10	Average Precision@15	MAP
TF-IDF	0.85*	0.79*	0.73*	0.81*
TextRank	0.87*	0.83	0.73*	0.83*
LexRank	0.85*	0.8*	0.72*	0.8*
Kim et al.	0.87*	0.81*	0.76*	0.84*
NERank <sub>Uni</sub>	0.80*	0.75*	0.71*	0.78*
NERank <sub><math>\alpha=0</math></sub>	0.72*	0.61*	0.51*	0.62*
NERank	<b>0.92</b>	<b>0.87</b>	<b>0.79</b>	<b>0.89</b>

The results are shown in Table 3. We can see our method outperforms baselines *TF-IDF*, *TextRank*, *LexRank* and *Kim et al.* and *TextRank* because these classical methods mostly capture the statistical characteristics of words and do not exploit the latent topics in document collections. The comparison between the variants and *NERank* shows that our topic rank function and meta-path constrained random walk algorithm are effective to boost the performance of ER. The results of paired t-test between *NERank* and baselines confirm that our method outperforms other approaches.

**Case Study.** We present the ER results of four events generated by our approach. Due to space limitation, we only present top-10 entities shown in Table 4. It can be seen that our approach can extract and rank entities from documents effectively.

**Table 4.** Top-10 entities of documents related to three events.

Entity	Egypt Revolution	Libya War	BP Oil Spill
1	Egypt	Libya	BP
2	Mohamed Morsi	Muammar Gaddafi	Gulf of Mexico
3	Hosni Mubarak	Tripoli	Barack Obama
4	Cario	NATO	Louisiana
5	Muslim Brotherhood	Benghazi	Coast Guard
6	Tahrir Square	Barack Obama	United States
7	Israel	Misrata	Tony Hayward
8	Middle East	United States	Deepwater Horizon
9	United States	National Transitional Council	Florida
10	Tunisia	Syria	Transocean

## 7 Conclusion and Future Work

In this paper, we formalize and address the problem of entity ranking. We design a TTG model to represent the semantic relations between documents, topics and entities. A meta-path constrained random walk algorithm is proposed to calculate the ranks of entities after estimating the prior ranks of topics by three quality metrics. The experimental results on two datasets demonstrate that the proposed approaches achieve accurate results. In the future, we will explore the task of joint entity linking and ranking for knowledge base population.

## Appendix: Mathematical Analysis of *NERank*

We prove that the random walk algorithm of *NERank* will converge and derive the close-form solution. Let  $\mathbf{T}_n$  denote the  $|T| \times 1$  matrix which represents the ranks of topics in the  $n^{th}$  iteration. Specially,  $\mathbf{T}_0$  is the prior rank matrix for topics. Let  $\mathbf{E}_n$  denote the  $|E| \times 1$  entity rank matrix in the  $n^{th}$  iteration. Based on the random walk process, the rank update of topics for TDT meta-path is formulated as:  $\mathbf{T}_n = \Theta_{\mathbf{R}}^T \Theta \cdot \mathbf{T}_{n-1}$  where  $\Theta_{\mathbf{R}}$  is the row-normalized matrix of  $\Theta$ . Similarly, for TET meta-path, we have  $\mathbf{T}_n = \hat{\Phi}_{\mathbf{C}} \hat{\Phi}_{\mathbf{R}}^T \cdot \mathbf{T}_{n-1}$  where  $\hat{\Phi}_{\mathbf{R}}$  and  $\hat{\Phi}_{\mathbf{C}}$  are the row-normalized and column-normalized matrices of  $\hat{\Phi}$ , respectively. The update rule in one iteration is formulated as:

$$\mathbf{T}_n = \alpha \cdot \Theta_{\mathbf{R}}^T \Theta \cdot \mathbf{T}_{n-1} + \beta \cdot \hat{\Phi}_{\mathbf{C}} \hat{\Phi}_{\mathbf{R}}^T \cdot \mathbf{T}_{n-1} + (1 - \alpha - \beta) \cdot \mathbf{T}_0$$

For simplicity, we define  $\mathbf{M} = \alpha \cdot \Theta_{\mathbf{R}}^T \Theta + \beta \cdot \hat{\Phi}_{\mathbf{C}} \hat{\Phi}_{\mathbf{R}}^T$ . By iteration, we have  $\mathbf{T}_n = \mathbf{M}^n \cdot \mathbf{T}_0 + (1 - \alpha - \beta) \cdot \sum_{i=0}^{n-1} \mathbf{M}^i \cdot \mathbf{T}_0$ . Because  $\lim_{n \rightarrow \infty} \mathbf{M}^n = \mathbf{0}$  and  $\lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} \mathbf{M}^i = (\mathbf{I} - \mathbf{M})^{-1}$ , the limit of matrix series  $\{\mathbf{T}_n\}$  is derived as:

$$\lim_{n \rightarrow \infty} \mathbf{T}_n = \lim_{n \rightarrow \infty} \mathbf{M}^n \cdot \mathbf{T}_0 + (1 - \alpha - \beta) \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} \mathbf{M}^i \cdot \mathbf{T}_0 = (1 - \alpha - \beta)(\mathbf{I} - \mathbf{M})^{-1} \mathbf{T}_0$$

where  $\mathbf{I}$  is the  $|T| \times |T|$  identity matrix. Therefore, the ranks of topics will converge in *NERank*. Because the rank of entities  $\mathbf{E}_n$  can be computed by  $\mathbf{E}_n = \hat{\Phi}_{\mathbf{R}}^T \cdot \mathbf{T}_n$ . Denote  $\mathbf{E}^*$  as the close form solution vector for entity ranks. We have

$$\mathbf{E}^* = (1 - \alpha - \beta) \cdot \hat{\Phi}_{\mathbf{R}}^T (\mathbf{I} - \alpha \cdot \Theta_{\mathbf{R}}^T \Theta - \beta \cdot \hat{\Phi}_{\mathbf{C}} \hat{\Phi}_{\mathbf{R}}^T)^{-1} \cdot \mathbf{T}_0$$

where the rank of entity  $e_i$  (i.e.,  $r(e_i)$ ) is the  $i^{th}$  element in  $\mathbf{E}^*$ .

## References

1. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Comput. Netw.* **30**(1-7), 107-117 (1998)
2. Ganesan, K., Zhai, C.: Opinion-based entity ranking. *Inf. Retr.* **15**(2), 116-150 (2013)

3. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: EMNLP, pp. 404–411 (2004)
4. Kaptein, R., Serdyukov, P., de Vries, A.P., Kamps, J.: Entity ranking using wikipedia as a pivot. In: CIKM, pp. 69–78 (2010)
5. de Vries, A.P., Vercoustre, A.-M., Thom, J.A., Craswell, N., Lalmas, M.: Overview of the INEX 2007 entity ranking track. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds.) INEX 2007. LNCS, vol. 4862, pp. 245–251. Springer, Heidelberg (2008)
6. Wang, C., Zhang, R., He, X., Zhou, A.: NERank: ranking named entities in document collections. In: WWW, pp. 123–124 (2016)
7. Balog, K., de Rijke, M.: Determining expert profiles (with an application to expert finding). In: IJCAI, pp. 2657–2662 (2007)
8. Nie, Z., Zhang, Y., Wen, J., Ma, W.: Object-level ranking: bringing order to web objects. In: WWW, pp. 567–574 (2005)
9. Lee, S., Song, S., Kahng, M., Lee, D., Lee, S.: Random walk based entity ranking on graph for multidimensional recommendation. In: RecSys, pp. 93–100 (2011)
10. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW, pp. 517–526 (2002)
11. Ilieva, E., Michel, S., Stupar, A.: The essence of knowledge (bases) through entity rankings. In: CIKM, pp. 1537–1540 (2013)
12. Meij, E., Weerkamp, W., de Rijke, M.: Adding semantics to microblog posts. In: WSDM, pp. 563–572 (2012)
13. Cornolti, M., Ferragina, P., Ciaramita, M.: A framework for benchmarking entity-annotation systems. In: WWW, pp. 249–260 (2013)
14. Usbeck, R., Röder, M., Ngomo, A.N., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B., Ferragina, P., Lemke, C., Moro, A., Navigli, R., Piccinno, F., Rizzo, G., Sack, H., Speck, R., Troncy, R., Waitelonis, J., Wesemann, L.: GERBIL: general entity annotator benchmarking framework. In: WWW, pp. 1133–1143 (2015)
15. Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating non-local information into information extraction systems by gibbs sampling. In: ACL (2005)
16. Jijkoun, V., Khalid, M.A., Marx, M., de Rijke, M.: Named entity normalization in user generated content. In: AND, pp. 23–30 (2008)
17. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
18. Shen, W., Wang, J., Luo, P., Wang, M.: LINDEN: linking named entities with knowledge base via semantic knowledge. In: WWW, pp. 449–458 (2012)
19. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.* **81**(1), 53–67 (2010)
20. Tran, G.B., Alrifai, M., Nguyen, D.Q.: Predicting relevant news events for timeline summaries. In: WWW, pp. 91–92 (2013)
21. Tran, G., Alrifai, M., Herder, E.: Timeline summarization from relevant headlines. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 245–256. Springer, Heidelberg (2015)
22. Zaragoza, H., Rode, H., Mika, P., Atserias, J., Ciaramita, M., Attardi, G.: Ranking very many typed entities on wikipedia. In: CIKM, pp. 1015–1018 (2007)
23. Erkan, G., Radev, D.R.: Lextrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)* **22**, 457–479 (2004)
24. Kim, Y., Kim, M., Cattle, A., Otmakhova, J., Park, S., Shin, H.: Applying graph-based keyword extraction to document retrieval. In: IJCNLP, pp. 864–868 (2013)