Front. Comput. Sci., 2018, 12(3): 504–517
https://doi.org/10.1007/s11704-017-6471-4

RESEARCH ARTICLE

NERank+: a graph-based approach for entity ranking in document collections

Chengyu WANG[1], Guomin ZHOU[2], Xiaofeng HE (✉)[1], Aoying ZHOU[3]

1 Shanghai Key Laboratory of Trustworthy Computing, School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China
2 Department of Computer and Information Technology, Zhejiang Police College, Hangzhou 310053, China
3 School of Data Science and Engineering, East China Normal University, Shanghai 200062, China

Abstract Most entity ranking research aims to retrieve a ranked list of entities from a Web corpus given a user query. The rank order of entities is determined by the relevance between the query and contexts of entities. However, entities can be ranked directly based on their relative importance in a document collection, independent of any queries. In this paper, we introduce an entity ranking algorithm named NERank+. Given a document collection, NERank+ first constructs a graph model called Topical Tripartite Graph, consisting of document, topic and entity nodes. We design separate ranking functions to compute the prior ranks of entities and topics, respectively. A meta-path constrained random walk algorithm is proposed to propagate prior entity and topic ranks based on the graph model. We evaluate NERank+ over real-life datasets and compare it with baselines. Experimental results illustrate the effectiveness of our approach.

Keywords entity ranking, Topical Tripartite Graph, prior rank estimation, meta-path constrained random walk

1 Introduction

Ranking problems have been extensively studied to bring order to varying types of objects to support Web applications, such as Web pages for search engines [1], commercial products for personalized recommendation [2], and textual units for keyword extraction [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 2007 and 2009, to rank entities from Web corpora given a query topic [4, 5]. The task of the WSDM 2016 Cup is to rank research articles based on Microsoft Academic Graph.

In traditional ER tasks, the rank order of entities is measured by the relevance between a query topic (e.g., *impressionist art in the Netherlands* in INEX [4]) and entities with contextual information, which is query-dependent. However, we observe that entities have an intrinsic rank order based on the relative importance in the documents. For example, in news articles reporting *Haiti Earthquake*, important entities should be key elements that are closely involved in the event, including people (e.g., Barack Obama), locations (e.g., Haiti, Port-au-Prince), organizations (e.g., United Nations, United States), etc. (Please refer to the background information of this event in the respective Wikipedia page). Thus, the goal of the ER problem addressed in this paper is to *rank multiple types of entities in a document collection based on the relative importance of entities*. Moreover, the task of ER is vital for several Web-scale applications, discussed as follows:

- Entity-oriented Web search It facilitates Web entity recommendation, rather than retrieve a list of Web documents that are relevant to the user query but contain abundant or irrelevant information.
- Web semantification It identifies important entities

Received September 27, 2016; accepted February 22, 2017

E-mail: xfhe@sei.ecnu.edu.cn

from Web documents and helps to 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 existing knowledge bases.

The challenge of ER is that the rank order of entities should be determined by the contents of the document collection, with no other knowledge sources or user queries available. Additionally, the importance of entities is expressed implicitly in the form of natural language text, which can not be measured or computed in a straight forward manner. Therefore, it is difficult to extend traditional ER techniques to solve the proposed task.

In this paper, we introduce a graph-based ranking algorithm named NERank+ to address this issue[1]. Given a document collection as input, we mine latent topics and model the semantic relations between documents, topics and entities in a graph model called *Topical Tripartite Graph* (TTG), which is a tripartite graph with edge weights. We design separate ranking functions to calculate the prior ranks of entities and topics. 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. We also prove the convergence of NERank+ and derive the close form solution of our algorithm.

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

- We introduce the problem of ER. A graph model TTG is proposed to represent the semantic relations between documents, topics and entities via topic modeling.
- We design separate ranking functions to calculate the prior ranks of entities and topics. A meta-path constrained random walk algorithm is proposed to compute the final 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 formally and introduce the general framework of NERank+ in Section 3. The proposed approach is described in Sections 4 and 5 in detail. Experimental results are presented in Section 6. We conclude our paper and discuss the future work in Section 7.

2 Related work

We divide the related work into two parts: the first summarizes methods in traditional ER research, and the second deals with keyword extraction from documents.

2.1 Traditional ER research

Research efforts on traditional ER have been put to address the problem of retrieving a ranked list of entities given a query. In the task of traditional 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 [4,5].

Besides these ER tracks, ER provides a paradigm to rank and retrieve information at an entity level in the field of Web search, rather than the document level. 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. [11] use the Wikipedia category structure as a pivot to identify key entities from Web documents. They reduce the problem of Web ER to Wikipedia ER. Ilieva et al. [12] make use of the rich attribute information in knowledge bases to improve the coverage and quality of ER. However, most existing work either focuses on query-dependent ER (such as ER tracks in INEX [11]) or specific applications such as personalized recommendation [2], Web search [8], etc. Therefore, these methods can not be employed to solve and evaluate the ER task in this paper.

2.2 Keyword extraction

Another thread of related work is keyword extraction, which generates a ranked order of words from documents. The ER task is similar to keyword extraction because of the similar ranking procedure and data sources.

[1] We name this algorithm NERank+ because it is an improved version of the algorithm NERank introduced in the earlier version of the paper presented in APWeb 2016 [6]

In the literature, TextRank [3] employs the PageRank algorithm [1] to calculate ranks of words or sentences in plain documents to support keyword extraction and document summarization. LexRank [13] computes relative importance of textual units based on the eigenvector centrality in a graph representation of documents. Zhang et al. [14] propose several intrinsic features and the relatedness measurements between words to improve the performance of keyword extraction. Kim et al. [15] integrates the semantic similarity between words into graph-based keyword extraction approaches to support document retrieval.

The accurate estimation of word similarity on the semantic level is beneficial to calculate the relative importance of words. Wang et al. [16] use WordNet as knowledge source to rank words based on PageRank. The similarity of words can be also computed based on the distributed representations of words. Wang et al. [17] find that the usage of word embeddings boosts the performance of keyword extraction in scientific publications. Besides keywords, Hofmann et al. [18] identify key phases considering the structure of a document. For short texts, Meij et al. [19] apply learning-to-rank models (such as Rank SVM models and gradient boosted regression trees) to extract key concepts in tweets.

The similarity between keyword extraction and ER is that both tasks aim to give a ranked order of a specific type of textual units. However, the focus of traditional keyword extraction research is to select important words (mostly verbs and nouns) in a single document. In contrast, our work pays more attention to the correlation between major topical events and key elements in a document collection. The topic coherence among key entities in different documents and the "unnormalization" issue of entities also require to be addressed.

3 Entity ranking problem

In this section, we present our ER problem formally, with important notations summarized in Table 1. Next, we discuss the general procedure of NERank+.

3.1 Problem statement

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 that appears in any 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 2. Both "United States" and "US" in news articles related to Haiti Earthquake refer to the country United States. If the two mentions are unnormalized, they will receive separate, inconsistent and under-estimated rank values (i.e., 0.12 and 0.1), rather than a single, uniform rank.

Table 1 Important notations

Notation	Description
D	The collection of input documents
E	The collection of normalized entities
$e \in E$	A normalized entities in E
$r(e)$	The rank of normalized entity e
M	The collection of entity mentions
$m \in M$	An entity mention in M
T	The collection of latent topics
$t \in T$	A topic t in T
W	The collection of common words
G_D	The TTG w.r.t. the document collection D
$\mathbf{\Theta}$	The document-topic distribution matrix
$\mathbf{\Phi}$	The topic-textual unit distribution matrix
$\hat{\mathbf{\Phi}}$	The topic-entity matrix
$r_0(t_i)$	The prior rank of topic t_i
$r_0(e_i)$	The prior rank of normalized entity e_i

Table 2 Comparison between unnormalized and normalized ranking

Unnormalized ranking		Normalized ranking	
Entity mention	Rank	Normalized entity	Rank
Haiti	0.35	Haiti	0.35
Port-au-Prince	0.25	Port-au-Prince	0.25
United States	0.12	United States	0.22
US	0.1	···	···

In this paper, before we compute the ranks of these entities, we employ a named entity normalization (NEN) procedure to map each entity mention $m \in M$ to its normalized form $e \in E$. We assign each entity $e \in E$ a rank $r(e)$ to represent the relative importance in D. We present the definition of ER as follows.

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) $0 \leqslant r(e) \leqslant 1$ and (2) $\sum_{e \in E} r(e) = 1$.

For the illustration purpose, the high-level process of ER is presented in Fig. 1. We also provide a simple example w.r.t. Haiti Earthquake to show the data processing steps of the proposed approach. The input is a collection of news articles related to Haiti Earthquake. Firstly, all the entities mentions M are recognized by an NE tagger, such as "US"

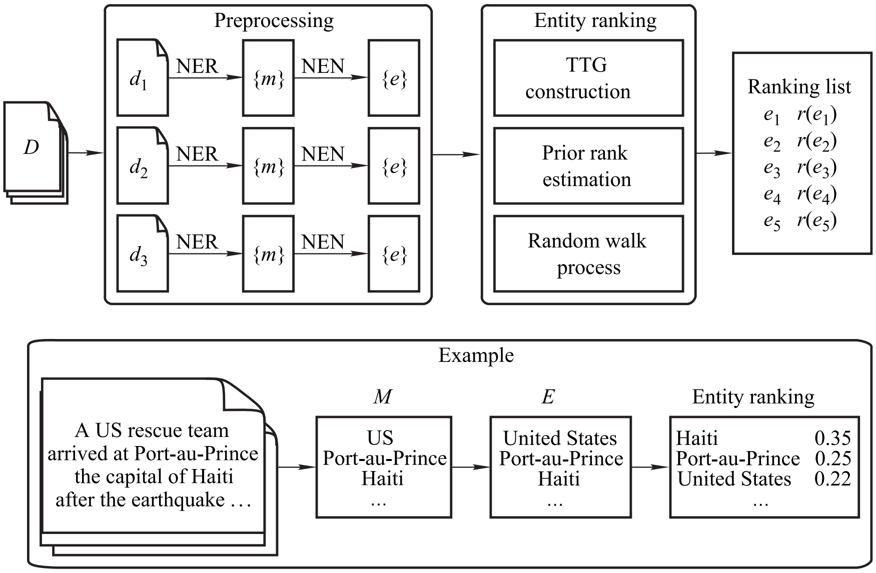

Fig. 1 Illustration of the ER process and a simple example

and “Haiti”. After that, these mentions are normalized to their referent entities E, e.g., “United States” and “Haiti”. Finally, the ranking order of entities is generated by three steps, i.e., TTG construction, prior rank estimation and random walk process.

The task definition of ER in this paper is similar to the task *Ranked-concepts to Wikipedia* (Rc2W) [20] and the more general task *Ranked-concepts to Knowledge Base* (Rc2KB) in the entity annotator benchmark GERBIL [21]. We notice that both tasks are comprised of two sub-steps: (i) entity linking (which maps an entity mention to an existing entity in a knowledge base) and (ii) entity ranking (which generates the ranked order of entities based on the contextual information). 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 in our paper. The ranked order of entities can be generated as long as they are recognized and normalized.

3.2 Major steps in NERank+

As shown in Fig. 1, the pre-processing steps of NERank+ are NER and NEN. NERank+ consists of three major steps: TTG construction, prior rank estimation and random walk process. In the first step, the tripartite graph model TTG G_D is constructed by estimating document-topic distributions $\mathbf{\Theta}$ and topic-textual unit distributions $\mathbf{\Phi}$ via entity-aware topic modeling. For example, we may extract a topic t_i related to the start of the revolution and another topic t_j related to the presidency of Mohamed Morsi.

After that, for each topic $t \in T$, we design a ranking function to estimate the prior rank $r_0(t)$ of topic t based on a linear combination of three quality metrics (i.e., prior probability, entity richness and topic specificity). This is used to indicate which of these topics are related to major aspects discussed in these news articles and which only provide some background information.

For each normalized entity $e \in E$, the prior rank $r_0(e)$ is calculated based on the statistical characteristics of entity e in the document collection D. The prior rank of an entity considers the relative importance of entities in a single article. Finally, a meta-path constrained random walk process is employed to compute the final rank $r(e)$ for each normalized entity $e \in E$, which combines all the factors mentioned previously.

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.

4.1 Topical Tripartite Graph

The TTG is a tripartite graph to model the semantic relations among document-topic and topic-entity pairs. An simple example of the TTG is illustrated in Fig. 2. 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.

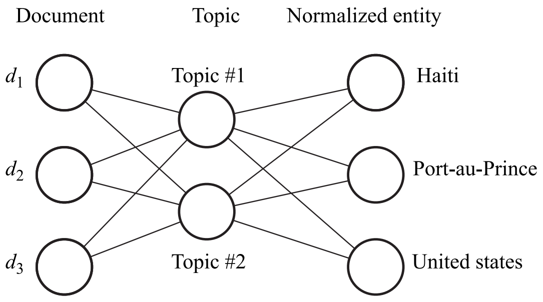

Fig. 2 Example of a TTG w.r.t. Haiti Earthquake

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 normalized entities E. R_{DT} and R_{TE} are edge sets that connect nodes between document-topic and topic-entity 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 $(t_i, e_j) \in R_{ET}$.

In Fig. 2, we can see that the graph structure of a TTG can model the relations between the news articles and normalized entities effectively, using topics as "bridges". The strength of the connections is calculated by entity-aware topic modeling, which will be introduced in the next subsection.

4.2 Graph construction

The TTG construction process includes two parts: (1) named entity recognition and normalization and (2) entity-aware topic modeling.

4.2.1 Named entity recognition and normalization

Entities in documents can be automatically recognized by the NER tagger, such as Conditional Random Fields [22]. Before we construct the TTG, entity normalization is necessary to transform entity mentions recognized by NER to normalized forms. In this paper, we employ the algorithm proposed by Jijkoun et al. [23] for entity normalization, which employs the techniques of approximate name matching, identification of missing references and name disambiguation. Due to space limitations, we omit the details here.

4.2.2 Entity aware topic modeling

Topic models such as LDA [24] 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 "Port-au-Prince is the capital of Haiti", we treat "Port-au-Prince" and "Haiti" as normalized entities, and "capital" as a common word.

After NER and NEN, LDA is employed to model the document-topic distributions $\mathbf{\Theta}$ (represented as a $|D|\times|T|$ matrix) and the topic-textual unit distributions $\mathbf{\Phi}$ ($|T| \times |E \cup W|$ matrix) given the document collection D. To generate these distributions, we model a text corpus as a collection of news articles and a news article as a collection of normalized entities and common words. We do not distinguish the differences between normalized entities and common words in this step and employ the Gibbs sampling algorithm [24] to calculate these distributions. The approaches to generate topic-word distributions in documents in the previous study and the topic-textual unit distributions in this paper are the same. After we obtain the distribution for each topic (i.e., $\mathbf{\Phi}$), we can identify which normalized entities and common words are heavily involved in this topic.

In Table 3, we present some topics we discovered in the collection of news articles w.r.t Haiti Earthquake. 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. For example, Topics #1 and #2 are obviously more important in Haiti Earthquake, compared to other topics.

The weights of edges 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 ith row and the jth column of $\mathbf{\Theta}$. Similarly, the semantic relations between topics and entities can be measured by the topic-textual unit distribution. We remove columns for topic-common word distributions in $\mathbf{\Phi}$, and denote the rest part of the matrix as $\hat{\mathbf{\Phi}}$ (called topic-entity matrix). For topic t_i and entity e_j, $w_{te}(t_i, e_j) = \hat{\phi}_{i,j}$ where $\hat{\phi}_{i,j}$

Table 3 Topics discovered in news articles w.r.t. Haiti Earthquake

Topic	Top normalized entities	Top common words	Description
#1	Haiti, United States, Port-au-Prince	earthquake, quake, people	The earthquake happened
#2	Haiti	days, rescue, alive, miracle	The rescue mission
#3	Haiti	hospital, patients, amputation	Doctors operated on the wounded
#4	-	raise, fundraising, donor	Fundraising activities
#5	-	program, media, discuss	Comments in media related to the earthquake

is the element in the ith row and the jth column of $\hat{\mathbf{\Phi}}$.

5 Entity ranking algorithm

In this section, we present the ranking algorithm by introducing ranking functions for the prior ranks of topics and entities. After that, a meta-path constrained random walk algorithm is proposed to calculate the ranks of entities by propagating prior ranks over the TTG. In this way, the semantic relations between entities and documents and prior knowledge about entities can be integrated in a unified model.

5.1 Prior entity rank estimation

As discussed in Section 2, we employ a variant of existing keyword extraction algorithm TextRank [3] to calculate the prior rank of entities, which is an unsupervised graph-based ranking model for text processing. In TextRank, a document is represented as an undirected graph where vertices are words. There exists an edge between two words if they are close enough in the document. In the implementation, we add an edge between two words if they appear in the same sentence. After that, the PageRank algorithm is employed to calculate the scores of words, which are called TextRank scores. Denote $r_d(e_i)$ as the TextRank score of entity e_i in document $d \in D$. If entity e_i does not exist in document d, we set $r_d(e_i) = 0$. The prior rank of entity e_i w.r.t. document collection D is defined as follows:

$$r_0(e_i) = \frac{1}{\sum_{j=1}^{|E|} r_0(e_j)} \sum_{d \in D} r_d(e_i).$$

By using the above approach, we encode the local evidence of entity importance into NERank+, without considering the topical coherence among ranks of entities in different documents.

5.2 Prior topic rank estimation

Entity aware topic modeling can provide prior knowledge about topics. For example, in Table 3, we can see that Topics #1 and #2 are directly about major events in Haiti Earthquake and Topics #3–#5 discuss different aspects related to Haiti Earthquake, but are less relevant. To facilitate ER, we design the following quality metrics and calculate the prior ranks of topics by a ranking function, introduced as follows.

5.2.1 Quality metrics

We present the definitions of the three quality metrics.

Quality metric 1 (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 3), while others are only related to a few articles (e.g., Topic #5). We define the prior probability $pr(t_i)$ of topic $t_i \in T$ using document-topic distributions as

$$pr(t_i) = \frac{1}{|D|} \sum_{j=1}^{|D|} \theta_{j,i}.$$

Because $\sum_{j=1}^{|D|} \sum_{t=1}^{|T|} \theta_{j,i} = |D|$, $|D|$ is served as a normalization factor for prior probability.

Quality metric 2 (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 constant.

Quality metric 3 (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 obverse 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 $\langle \theta_{1,i}, \theta_{2,i}, \ldots, \theta_{|D|,i} \rangle$. Similar to entropy, the unnormalized "specificity" of topic t_i can be computed as

$$\tilde{ts}(t_i) = \sum_{j=1}^{|D|} \theta_{j,i} \log_2 \theta_{j,i}.$$

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 ϵ, we set $ts(t_i) = 0$. Hence, the topic specificity of t_i is defined as:

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

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

5.2.2 Ranking function

Combining the three quality metrics together, we can generate a feature vector for each topic $t_i \in T$, i.e., $\vec{F}(t_i) = \langle pr(t_i), er(t_i), ts(t_i) \rangle$. Denote $\vec{W}$ as the weight vector where each element in $\vec{W}$ 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 $r_0(t_i) = \vec{W}^T \cdot \vec{F}(t_i)$.

To learn the weights $\vec{W}$ for the features, we employ the max-margin technique introduced in [25]. Given two topics t_i and t_j and their respective top common words and normalized entities, 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:

$$\vec{W}^T \cdot \vec{F}(t_i) - \vec{W}^T \cdot \vec{F}(t_j) \geqslant 1 - \xi_{i,j},$$

where $\xi_{i,j} \geqslant 0$ is a slack variable. This learning problem can be modeled as training a linear SVM classifier with the objective function $||\vec{W}||_2^2 + C \cdot \sum_{i,j} \xi_{i,j}$, where C is a tolerance parameter.

5.2.3 Discussion

An remaining issue related to the three quality metrics is that these metrics are not necessarily statistically independent. In Table 4, we present the pairwise Pearson correlation between these quality metrics based on our dataset TimelineData (see Section 6). We can see that the positive correlation does exist among these metrics. However, this issue does not in fact harm the performance of NERank+. This is because we employ an SVM based approach to determine which topics are “good” and which are “bad”. The features (i.e., quality metrics in this case) do not need to be un-related since we do not try to model the generation process of the data. Additionally, our experiments show that adding more quality metrics can improve the performance of NERank+.

Table 4 Correlation between three quality metrics

	Metric 1	Metric 2	Metric 3
Metric 1	1	0.7813	0.8134
Metric 2	0.7813	1	0.7647
Metric 3	0.8134	0.7647	1

5.3 Meta-path constrained random walk algorithm

With prior ranks of topics and entities estimated, we aim to propagate prior ranks to other nodes in order to obtain final entity ranks by considering the correlation among documents, topics and entities.

In a TTG, we observe that only topic nodes are connected with all other types of nodes (i.e., documents and entities). Thus, we define topic-centric meta-paths to constrain the behavior of random walkers. Denote $x \to y$ 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 $t_i \to d_j \to t_k$ 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 $t_i \to e_j \to t_k$ 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 [26], we compute the ranks of entities by meta-path constrained random walk. To better fit the graph structure of a TTG, 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, or jumps entity or topic nodes with the probability proportional to the respective prior ranks. Denote α, β and γ as random walk parameters where $\alpha > 0$, $\beta > 0$, $\alpha + \beta < 1$ and $0 < \gamma < 1$. One iteration of the random walk process is shown as follows:

- **Choice 1** With probability α, the random surfer walks through a TDT meta-path $t_i \rightarrow d_j \rightarrow t_k$. d_j is selected with probability $\theta_{j,i}/\sum_{d_k \in D} \theta_{k,i}$ for all $d_j \in D$. Next, t_k is selected with probability $\theta_{j,k}$ for all $t_k \in T$.
- **Choice 2** With probability $\beta\gamma$, the random surfer walks through a TET meta-path $t_i \rightarrow e_j \rightarrow t_k$. e_j is selected with probability $\hat{\phi}_{i,j}/\sum_{e_k \in E} \hat{\phi}_{i,k}$ for all $e_j \in E$. Next, t_k is selected with probability $\hat{\phi}_{k,j}/\sum_{t_m \in T} \hat{\phi}_{m,j}$ for all $t_k \in T$.
- **Choice 3** With probability $\beta(1 - \gamma)$, the random surfer walks through a TET meta-path $t_i \rightarrow e_j \rightarrow t_k$. e_j is selected with probability $r_0(e_j)$ for all $e_j \in E$. Next, t_k is selected with probability $\hat{\phi}_{k,j}/\sum_{t_m \in T} \hat{\phi}_{m,j}$ for all $t_k \in T$.
- **Choice 4** 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 $r(e_i) = s(e_i)/\sum_{e_j \in E} s(e_j)$.

We present the pseudo code for the implementation of the meta-path constrained random walk algorithm in the following. It begins with the initialization of counters for documents, topics and entities. After that, the meta-path constrained random walk processes iterates until the ranks of entities converge[2]. Finally, a collection of ⟨entity, rank⟩ pairs are returned.

Because the meta-path constrained random walk algorithm is a random algorithm, it is difficult to qualify the complexity of the algorithm. In an ER task, the number of topics is set as a constant. Thus, the runtime complexity is $O(k|D||E|)$ where k is the number of iterations. While the number of iterations k can not be determined beforehand, in Section 4, we prove that this algorithm has a close form solution. Therefore, this algorithm can be also implemented as matrix computation. In the experiments, we can see that it takes less than ten seconds to calculate the entity ranks in our test set. As for the space, we only need to store the prior ranks of entities and topics, the document-topic distribution matrix $\boldsymbol{\Theta}$ and the topic-entity matrix $\hat{\boldsymbol{\Phi}}$, together with the counters for all the nodes in the TTG. Thus, the space complexity is $O(|D||E|)$. Therefore, our algorithm is highly efficient for both running time and memory consumption.

Algorithm Meta-path constrained random walk algorithm

```
// counter initialization
for each $d_i \in D$ do
    $n(d_i) = 0$;
end for
for each $t_i \in T$ do
    $n(t_i) = 0$;
end for
for each $e_i \in E$ do
    $n(e_i) = 0$;
end for
// random walk process
Select $t_i \in T$ as the starting point with prob. $r_0(t_i)$;
while not converge do
    Generate random number: $r = Random(0, 1)$;
    if $r < \alpha$ then
        Generate a TDT meta-path $t_i \rightarrow d_j \rightarrow t_k$ based on Choice 1;
        $n(t_i) = n(t_i) + 1, n(d_j) = n(d_j) + 1, n(t_k) = n(t_k) + 1$;
    else if $r < \alpha + \beta\gamma$ then
        Generate a TET meta-path $t_i \rightarrow e_j \rightarrow t_k$ based on Choice 2;
        $n(t_i) = n(t_i) + 1, n(e_j) = n(e_j) + 1, n(t_k) = n(t_k) + 1$;
    else if $r < \alpha + \beta$ then
        Generate a TET meta-path $t_i \rightarrow e_j \rightarrow t_k$ based on Choice 3;
        $n(t_i) = n(t_i) + 1, n(e_j) = n(e_j) + 1, n(t_k) = n(t_k) + 1$;
    else
        Jump to $t_i \in T$ with prob. $r_0(t_i)$;
        $n(t_i) = n(t_i) + 1$;
    end if
end while
// entity rank calculation
for each $e_i \in E$ do
    $r(e_i) = \frac{s(e_i)}{\sum_{e_j \in E} s(e_j)}$;
end for
return Pairs of entity ranks $R = \{\langle e_i, r(e_i)\rangle | e_i \in E\}$;
```

5.4 Close form solution

We prove that the random walk algorithm of NERank+ will converge after a sufficient number of iterations, and derive the close-form solution of NERank+.

Let $\mathbf{T_n}$ denote the $|T| \times 1$ matrix which represents the ranks of topics in the nth iteration. Specially, $\mathbf{T_0}$ is the prior rank matrix of topics. Let $\mathbf{E_n}$ denote the $|E| \times 1$ entity rank matrix in the nth iteration. $\mathbf{E_0}$ is the prior rank matrix of entities. Based on the random walk process, the rank update of topics for TDT meta-path is formulated as: $\mathbf{T_n} = \boldsymbol{\Theta}_\mathbf{R}^\mathrm{T}\boldsymbol{\Theta} \cdot \mathbf{T_{n-1}}$ where $\boldsymbol{\Theta}_\mathbf{R}$ is the row-normalized matrix of $\boldsymbol{\Theta}$. Similarly, for TET meta-path, we have $\mathbf{T_n} = \hat{\boldsymbol{\Phi}}_\mathbf{C}\hat{\boldsymbol{\Phi}}_\mathbf{R}^\mathrm{T} \cdot \mathbf{T_{n-1}}$ where $\hat{\boldsymbol{\Phi}}_\mathbf{R}$ and $\hat{\boldsymbol{\Phi}}_\mathbf{C}$ are the row-normalized and column-normalized matrices

[2] In the implementation, the ranks of entities converge if the l_2 norm of the rank vector offset in two iterations is smaller than 0.01

of $\hat{\mathbf{\Phi}}$, respectively.

Based on the random walk process in one iteration, the update rule of topic ranks is formulated in a recurrent form:

$$\mathbf{T_n} = \alpha\cdot\mathbf{\Theta}_\mathbf{R}^\mathrm{T}\mathbf{\Theta}\cdot\mathbf{T_{n-1}}+\beta\cdot\hat{\mathbf{\Phi}}_\mathrm{C}(\gamma\hat{\mathbf{\Phi}}_\mathbf{R}^\mathrm{T}\cdot\mathbf{T_{n-1}}+(1-\gamma)\mathbf{E_0})+(1-\alpha-\beta)\cdot\mathbf{T_0}.$$

For simplicity, we define $\mathbf{M} = \alpha \cdot \mathbf{\Theta}_\mathbf{R}^\mathrm{T}\mathbf{\Theta} + \beta\gamma \cdot \hat{\mathbf{\Phi}}_\mathrm{C}\hat{\mathbf{\Phi}}_\mathbf{R}^\mathrm{T}$ and $\mathbf{C} = \beta(1-\gamma)\cdot\hat{\mathbf{\Phi}}_\mathrm{C}\mathbf{E_0} + (1-\alpha-\beta)\cdot\mathbf{T_0}$. Therefore, the update rule is: $\mathbf{T_n} = \mathbf{M}\cdot\mathbf{T_{n-1}} + \mathbf{C}$. Thus the non-iteration form of the update rule is: $\mathbf{T_n} = \mathbf{M}^n \cdot \mathbf{T_0} + \sum_{i=0}^{n-1}\mathbf{M}^i \cdot \mathbf{C}$.

Consider the l_1-norm of matrix $\mathbf{M}$:

$$||\mathbf{M}||_1 \leqslant \alpha \cdot ||\mathbf{\Theta}_\mathbf{R}^\mathrm{T}\mathbf{\Theta}||_1 + \beta\gamma \cdot ||\hat{\mathbf{\Phi}}_\mathrm{C}\hat{\mathbf{\Phi}}_\mathbf{R}^\mathrm{T}||_1.$$

Because $\mathbf{\Theta}_\mathbf{R}^\mathrm{T}\mathbf{\Theta}$ and $\hat{\mathbf{\Phi}}_\mathrm{C}\mathbf{\Phi}_\mathbf{R}^\mathrm{T}$ are again transition matrices, we have $||\mathbf{\Theta}_\mathbf{R}^\mathrm{T}\mathbf{\Theta}||_1 = 1$ and $||\hat{\mathbf{\Phi}}_\mathrm{C}\mathbf{\Phi}_\mathbf{R}^\mathrm{T}||_1 = 1$. Thus, $||\mathbf{M}||_1 \leqslant \alpha+\beta\gamma < \alpha + \beta < 1$. Denote $\rho(\mathbf{M})$ as the spectral radius of $\mathbf{M}$. Because $\rho(\mathbf{M}) \leqslant ||\mathbf{M}||_1 < 1$, the convergence of the power sequence of $\mathbf{M}$ is stated as $\lim_{n\to\infty}\mathbf{M}^n = \mathbf{0}$.

Let $\mathbf{I}$ be the $|T| \times |T|$ identity matrix. Then we have $\lim_{n\to\infty}\sum_{i=0}^{n-1}\mathbf{M}^i = (\mathbf{I}-\mathbf{M})^{-1}$. The limit of matrix series $\{\mathbf{T_n}\}$ is derived as:

$$\lim_{n\to\infty}\mathbf{T_n} = \lim_{n\to\infty}\mathbf{M}^n \cdot \mathbf{T_0} + \lim_{n\to\infty}\sum_{i=0}^{n-1}\mathbf{M}^i \cdot \mathbf{C} = (\mathbf{I}-\mathbf{M})^{-1}\mathbf{C},$$

which means the ranks of topics will converge in NERank+. Therefore, the close form solution of topic rank vector $\mathbf{T}^*$ is:

$$\mathbf{T}^* = (\mathbf{I}-\alpha\cdot\mathbf{\Theta}_\mathbf{R}^\mathrm{T}\mathbf{\Theta}-\beta\gamma\cdot\hat{\mathbf{\Phi}}_\mathrm{C}\mathbf{\Phi}_\mathbf{R}^\mathrm{T})^{-1}\cdot(\beta(1-\gamma)\cdot\hat{\mathbf{\Phi}}_\mathrm{C}\mathbf{E_0}+(1-\alpha-\beta)\cdot\mathbf{T_0}).$$

The rank of entities $\mathbf{E_n}$ can be derived by the rank of topics: $\mathbf{E_n} = \gamma\mathbf{\Phi}_\mathbf{R}^\mathrm{T} \cdot \mathbf{T_n} + (1-\gamma)\mathbf{E_0}$. Denote $\mathbf{E}^*$ as the close form solution of entity rank vector. We have

$$\begin{aligned}\mathbf{E}^* = \gamma\mathbf{\Phi}_\mathbf{R}^\mathrm{T} &\cdot (\mathbf{I} - \alpha \cdot \mathbf{\Theta}_\mathbf{R}^\mathrm{T}\mathbf{\Theta} - \beta\gamma \cdot \hat{\mathbf{\Phi}}_\mathrm{C}\mathbf{\Phi}_\mathbf{R}^\mathrm{T})^{-1} \cdot (\beta(1-\gamma)\cdot\hat{\mathbf{\Phi}}_\mathrm{C}\mathbf{E_0} \\ &+(1-\alpha-\beta)\cdot\mathbf{T_0}) + (1-\gamma)\mathbf{E_0},\end{aligned}$$

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

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 and present case studies to make the convincing conclusion.

6.1 Datasets and experimental settings

We use three newswire datasets in our experiments. Two datasets (i.e., TimelineData [27] and CrisisData [28]) are English news collections which have been employed in previous research. Because the number of events in these datasets is relatively small, we addtionally use our own Chinese news dataset for evaluation (i.e., EduData), described as follows:

- TimelineData The dataset has 4,650 news articles that are related to 17 international events, such as BP Oil Spill, Iraq War, etc. Each group of news articles belongs to a news agency, such as BBC, CNN, etc.
- CrisisData The dataset contains 15,534 news articles that report four armed conflicts. These articles are published by 24 news agencies, obtained using Google search engine.
- EduData The dataset contains 2,041 news articles that report 39 popular educational events in China in 2015, including Cheating in College Entrance Exam, Fudan Poisoning Case, etc. These articles are crawled from famous Chinese news websites, such as Sina.com, 163.com, etc.

To generate document collections, for English news datasets, 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. Besides, we use all 39 document collections in EduData. We conduct separate experiments on all document collections in the experiments.

In the following, we illustrate the detailed statistics of these news articles. In Fig. 3, we present the distribution of the number of sentences per news article. In average, there are 26.4 sentences per news article. Most news articles (67.3%) have 10–50 sentences. We also analyze the distribution of the number of normalized entities per news article, of which the results are shown in Fig. 4. Although approximately 4.8% of all the news articles have no named entities, the average number of normalized entities per news article is 6.4.

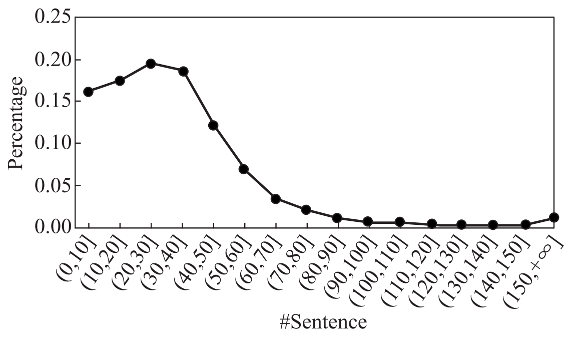

Fig. 3 Distribution of the number of sentences per news article

In the implementation, all the codes are written in JAVA. The experiments are conducted on a single machine with

2.9GHz CPU and 16GB memory. We use the open-source software *JGibbLDA* to estimate the parameters in topic modeling, *Stanford Named Entity Tagger* for English NER and *Ansj* for Chinese NER.

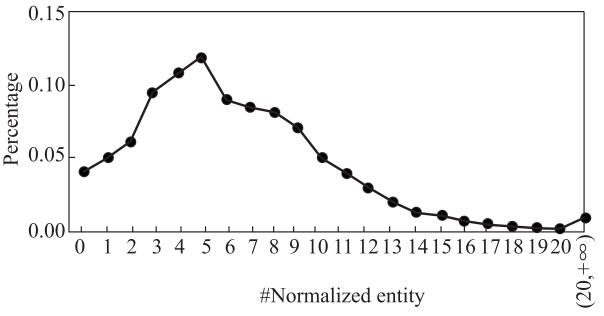

Fig. 4 Distribution of the number of normalized entities per news articles

6.2 Evaluation method

To our knowledge, there is no existing standard benchmark to evaluate the performance of ER addressed in this paper. Here, we introduce our ground truth acquisition method and evaluation metrics for ER.

6.2.1 Ground truth acquisition

For ground truth, we first obtain the English news summaries of each document collection from [27, 28], which are manually created by professional journalists. Based on the event summaries, we recruit a group of CS graduates to label normalized entities into four classes: "most important", "important", "related" and "unrelated". Following the evaluation framework in [29], we calculate the average score among all human labelers for each normalized entity based on the rank score table in Table 5. We finally have a ranked list of 15 normalized entities w.r.t. a document collection, which are regarded as "key" entities.

Table 5 Rank score table for ground truth acquisition

Rank level	Score	Example entities w.r.t. Haiti Earthquake
Most important	3	Haiti, Port-au-Prince
Important	2	United Nations, United States
Related	1	Bill Clinton, France
Unrelated	0	BBC

6.2.2 Evaluation metrics

To evaluate different algorithms for ER, we compare the top-k entities generated by machines with the ground truth ranking list. 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 (Avg P@K) and MAP in this paper.

To compare NERank+ with baselines, we additionally use *paired t-test* to evaluate the level of statistical significance. It is a special case of one-sample t-test to test the null hypothesis that the difference between two measurements is equal to zero. Let $f_1, f_2, \ldots, f_n$ be the respective ER performance w.r.t. n document collections using a baseline method under a certain evaluation metric, and $f_1^*, f_2^*, \ldots, f_n^*$ be the ER performance using NERank+ under the same evaluation metric. The sample mean and the standard deviation are as: $\bar{f} = \frac{1}{n}\sum_{i=1}^{n}(f_i - f_i^*)$ and $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(f_i - f_i^* - \bar{f})^2}$. The paired t-test uses $t_s = \frac{\bar{f}}{s/\sqrt{n}}$ as the test statistic and reports the p-value, indicating whether NERank+ outperforms the baseline with statistical significance.

6.3 Experimental results and analysis

In this subsection, we study the effectiveness of our model NERank+ under different configurations, and compare them with competitive baselines. For parameter analysis, we use five document collections from TimelineData and five from CrisisData as the development set to show how the performance is affected by the parameters. The rest of document collections are employed as the test set for comparison with baselines.

6.3.1 Parameter analysis

We analyze how the settings of parameters (i.e., the number of topics in topic modeling $|T|$ and parameters in the random walk process α, β and γ) in NERank+ can effect the performance of ER. We present the experimental results when we vary only one parameter at each time.

Because in NERank+, both prior ranks of entities and topics are embedded in the model, we first ignore the effects of prior topic ranks and set $\gamma = 0$. In Fig. 5, we fix $\alpha = \beta = 0.4$ and and change the number of topics of the topic model. 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 when the topic number is not too extreme. This is because when the topic number is too small, named entities and common words that are related to different aspects are likely to be "merged" into a single topic. When it is too large, each topic may carry little semantic meaning. In Fig. 5, we find that our approach achieves the highest performance with $|T| = 10$.

In Figs. 6 and 7, we set $|T| = 10$ and one parameter (α or β) to be 0.4 and vary the other. It shows that our algorithm is also not sensitive to the change of parameters α or

β. Therefore, as long as the parameters $|T|$, α and β are not set to extreme values, NERank+ can achieve high and relatively stable performance. Note that the weight vector $\vec{W}$ 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.

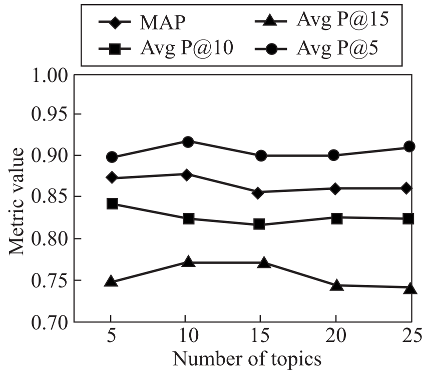

Fig. 5 Performance of NERank+ varying number of topics

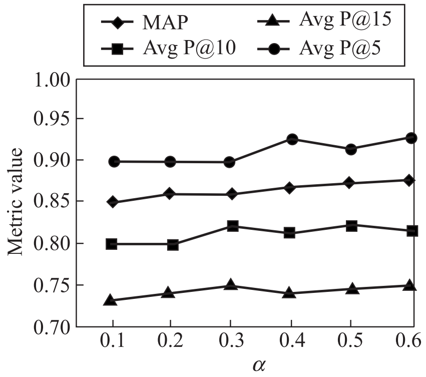

Fig. 6 Performance of NERank+ varying α

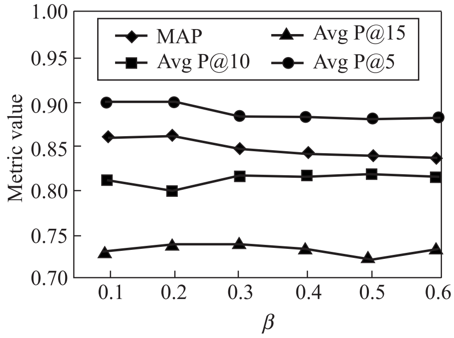

Fig. 7 Performance of NERank+ varying β

Next, we focus on evaluating the effectiveness of prior entity ranks by varying the value of γ from 0.1 to 0.9, with experimental results shown in Fig. 8. Intuitively, a larger γ will increase the importance of the prior entity ranks in NRank+; and at the same time decrease the importance of the topical coherence of ER between different topics and documents. Figure 8 shows this trade-off, indicating our method can achieve the best performance when the value of γ is 0.3.

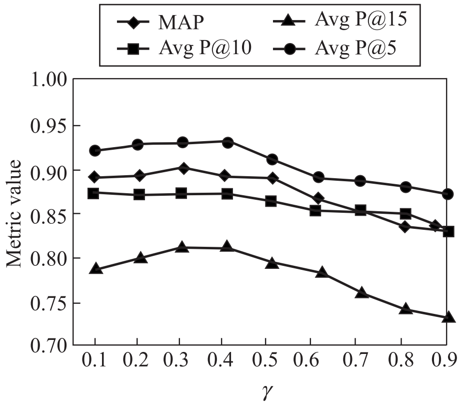

Fig. 8 Performance of NERank+ varying γ

6.3.2 Comparison with baselines

To our knowledge, there is no prior work concerning ranking entities directly from document collections. In this paper, we take simple ranking approaches, keyword extraction methods and variants of NERank+ as baselines:

- Frequency It ranks normalized entities based on the frequencies in the document collections.
- TextRank [3] It employs the graph-based ranking algorithm TextRank to generate a ranked list of words and entities, and next filters out common words. This method is equivalent of using prior entity ranks only.
- LexRank [13] It constructs a lexical centrality matrix based on word similarity and selects top-k entities based on the eigenvectors of the matrix.
- Kim et al. [15] It is a keyword extraction algorithm based on semantic similarity between words.
- NERank$_{Uni}$ It is the variant of our approach which sets prior topic ranks uniformly.
- NERank$_{\alpha=0}$ It is the variant of our approach which sets $\alpha = 0$ in random walk and thus ignores the semantic relatedness between documents and topics.
- NERank [6] It is the full implementation of the algorithm proposed in our conference paper [6].

The results are shown in Table 6. We can see our method outperforms baselines Frequency, TextRank [3], LexRank [13] and Kim et al. [15]. We believe this is because these classical methods mostly capture the statistical characteristics of words and do not exploit the latent topics in document collections. For example, based on Fig. 4, approximately 4.8% of news articles have no named entities, and thus do not provide any evidence of entity ranks in these methods. In contrast, NERank+ considers the global coherence of ER by estimating prior topic ranks.

Table 6 Evaluation results of different methods

Method	Avg P@5	Avg P@10	Avg P@15	MAP
Frequency	0.82*	0.76*	0.71*	0.77*
TextRank	0.85*	0.81	0.71*	0.79*
LexRank	0.83*	0.79*	0.71*	0.78*
Kim et al.	0.86	0.78*	0.75*	0.83
NERank$_{Uni}$	0.78*	0.74*	0.70*	0.76*
NERank$_{\alpha=0}$	0.68*	0.60*	0.69*	0.61*
NERank	**0.91**	**0.83**	**0.76**	**0.87**
NERank+	**0.92**	**0.86**	**0.80**	**0.88**

Note: $\star$: p-value$\leqslant$0.05

The comparison between the variants and NERank+ shows that our prior entity and topic rank estimation approaches and meta-path constrained random walk algorithm are effective to boost the performance of ER. The results of paired *t*-test between NERank+ and six baseline methods (i.e., Frequency, TextRank, LexRank, Kim et al., NERank$_{Uni}$ and NERank$_{\alpha=0}$) confirm that our method outperforms these approaches significantly with the confidence level of 95%. The performance of NERank+ has slight improvement over our prior work NERank in [6].

6.3.3 Effectiveness of topic quality metrics

One major contribution of NERank+ is that it can compute the "goodness" of topics by using three quality metrics. Table 7 illustrates the values of these metrics of five topics w.r.t. Haiti Earthquake (see Table 3). From the results, we can see that topics related to major aspects (i.e., Topics #1 and #2) have high values of all three metrics, compared to the other two topics. This means, these metrics are effective to distinguish meaningful or background topics, which is consistent with our intuition.

Table 7 Values of three quality metrics w.r.t. topics in Table 3

Topic	Metric 1	Metric 2	Metric 3
#1	0.231	0.187	0.256
#2	0.258	0.146	0.232
#3	0.169	0.112	0.134
#4	0.091	0.043	0.043
#5	0.034	0.021	0.012

In order to demonstrate which quality metrics are more effective, we evaluate the precision performance of our method NERank+ with different quality metric sets. The experimental results are shown in Fig. 9. We can see that NERank+ can obtain the best result with all metrics combined. Furthermore, the precision achieved by our method that employs entity richness and/or topic specificity is higher than the precision obtained by NERank+ using prior probability. This means, entity richness and topic specificity are more important and effective for the ER process.

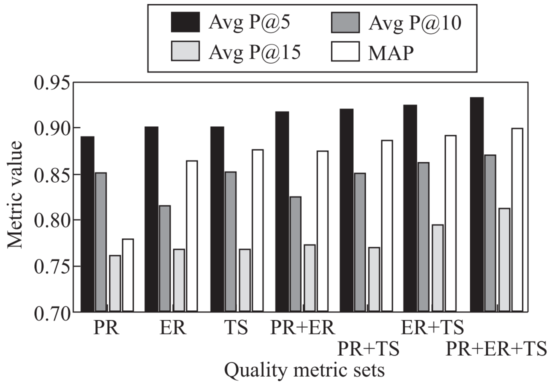

Fig. 9 Performance of NERank+ with different quality metric sets (Prior probability, entity richness and topic specificity are abbreviated as PR, ER and TS, respectively)

6.3.4 Efficiency performance

We evaluate the efficiency performance of our approach and compare it with other methods. Figure 10 illustrates the CPU time required for retrieving top-10 entities from document collections. We only list the efficiency performance for four events in CrisisData which have largest number of distinct entities. For fair comparison, all the pre-processing steps and offline modeling training have been done before running different ranking algorithms. Frequency is the most naive approach and thus the running time is the smallest. Our approach NERank+ is slightly slower than TextRank because TextRank is regarded as a module in NERank+. For all the four events, NERank+ use less than ten seconds. LexRank and Kim et al. are slower than NERank+, using over ten seconds in most cases.

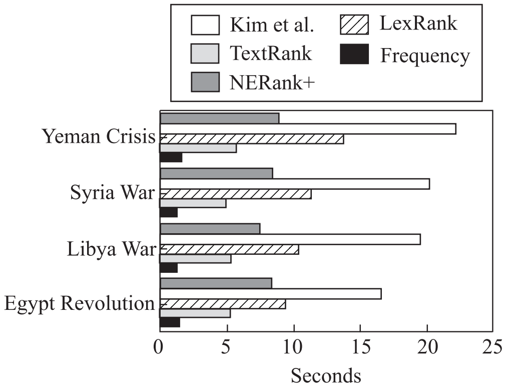

Fig. 10 Efficiency comparison between different methods

6.3.5 Case study on real-life events

We present the ER results of five events generated from their respective news articles. Due to space limitation, we only present top-10 entities of each event, shown in Table 8. Take the case "BP Oil Spill" as an example. The top-10 entities includes key elements involved in the BP Oil Spill disaster happened in 2010, including the BP former CEO Tony Hayward, major locations where the disaster took place such as Gulf of Mexico, the company BP and other parties that were involved in the event (e.g., Coast Guard). It can be seen that our approach can extract and rank entities from a collection of documents effectively.

Table 8 Top-10 entities of news articles related to five international events

Event	BP Oil Spill	Iraq War	Financial Crisis	Death of Michael Jackson	Haiti Earthquake
1	BP	Iraq	Barack Obama	Michael Jackson	Haiti
2	Gulf of Mexico	Saddam Hussein	United States	Conrad Murray	Port-au-Prince
3	Barack Obama	United States	George Bush	Los Angeles	United States
4	Louisiana	George Bush	China	Jermaine Jackson	United Nations
5	Coast Guard	United Kingdom	Washington	United States	European Union
6	United States	Tony Blair	Wall Street	AEG	Red Cross
7	Tony Hayward	Baghdad	Federal Reserve System	California	Caribbean
8	Deepwater Horizon	Basra	International Monetary Fund	Leona Lewis	Barack Obama
9	Florida	United Nations	Ben Bernanke	Justin Bieber	Wyclef Jean
10	Transocean	Europe	Europe	Edward Chernoff	Disasters Emergency Committee

7 Conclusion and future work

In this paper, we propose and solve the problem of ER. 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 entities and topics. The experimental results on two datasets demonstrate that the proposed approaches outperform several competitive baselines and achieve accurate results.

There are two pieces of future work. As discussed in Section 3, entity linking and ranking can be combined into a joint task so that the performance of both tasks can be mutually reinforced. Another piece of future work is related to other data sources. Currently, NERank+ only works with plain document collections. With proper extension, we can rank entities in other types of data sources such as tweets and Web pages.

Acknowledgements This work was partially supported by the National Key Research and Development Program of China (2016YFB1000904), Shanghai Agriculture Applied Technology Development Program, China (T20150302) and NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1509219). Chengyu Wang would like to thank the East China Normal University Outstanding Doctoral Dissertation Cultivation Plan of Action (YB2016040) for the support of his research. An earlier version of this paper "NERank: Bringing Order to Named Entities from Texts" was presented at the 18th Asia-Pacific Web Conference.

References

1. Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN System, 1998, 30(1–7): 107–117
2. Ganesan K, Zhai C X. Opinion-based entity ranking. Information Retrieval, 2013, 15(2): 116–150
3. Mihalcea R, Tarau P. Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. 2004, 404–411
4. Vries A, Vercoustre A, Thom J, Craswell N, Lalmas M. Overview of the INEX 2007 entity ranking track. In: Proceedings of the 6th International Workshop of the Initiative for the Evaluation of XML Retrieval. 2007, 245–251
5. Balog K, Vries A, Serdyukov P, Thomas P, Westerveld T. Overview of the TREC 2009 entity track. In: Proceedings of the 18th Text REtrieval Conference. 2009, 245–251
6. Wang C Y, Zhang R, He X F, Zhou G M, Zhou A Y. NERank: bringing order to named entities from texts. In: Proceedings of the 18th Asia-Pacific Web Conference. 2016, 1–13
7. Balog K, Rijke M. Determining expert profiles (with an application to expert finding). In: Proceedings of the 20th International Joint Conference on Artificial Intelligence. 2007, 2657–2662
8. Nie Z Q, Zhang Y Z, Wen J R, Ma W Y. Object-level ranking: bringing order to Web objects. In: Proceedings of the 14th International Conference on World Wide Web. 2005, 567–574
9. Lee S, Song S, Kahng M, Lee D, Lee S. Random walk based entity ranking on graph for multidimensional recommendation. In: Proceedings of ACM Conference on Recommender Systems. 2011, 93–100
10. Haveliwala T. Topic-sensitive pagerank. In: Proceedings of the 11th International World Wide Web Conference. 2002, 517–526
11. Kaptein R, Serdyukov P, Vries A, Kamps J. Entity ranking using Wikipedia as a pivot. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management. 2010, 69–78
12. Ilieva E, Michel S, Stupar A. The essence of knowledge (bases) through entity rankings. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management. 2013, 1537–1540

13. Erkan G, Radev D. Lexrank: graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, 2004, 22: 457–479

14. Zhang W, Feng W, Wang J Y. Integrating semantic relatedness and words' intrinsic features for keyword extraction. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence. 2013, 2225–2231

15. Kim Y, Kim M, Cattle A, Otmakhova J, Park S, Shin H. Applying graph-based keyword extraction to document retrieval. In: Proceedings of the 6th International Joint Conference on Natural Language Processing. 2013, 864–868

16. Wang J H, Liu J Y, Wang C. Keyword extraction based on pagerank. In: Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining. 2007, 857–864

17. Wang R, Liu W, McDonald C. Using word embeddings to enhance keyword identification for scientific publications. In: Proceedings of the 26th Australasian Database Conference. 2015, 257–268

18. Hofmann K, Tsagkias M, Meij E, Rijke M. The impact of document structure on keyphrase extraction. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. 2009, 1725–1728

19. Meij E, Weerkamp W, Rijke M. Adding semantics to microblog posts. In: Proceedings of the 5th International Conference on Web Search and Web Data Mining. 2012, 563–572

20. Cornolti M, Ferragina P, Ciaramita M. A framework for benchmarking entity-annotation systems. In: Proceedings of the 22nd International World Wide Web Conference. 2013, 249–260

21. Usbeck R, Röder M, Ngomo A, 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: Proceedings of the 24th International Conference on World Wide Web. 2015, 1133–1143

22. Finkel J, Grenager T, Manning C. Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics. 2005, 363–370

23. Jijkoun V, Khalid M, Marx M, Rijke M. Named entity normalization in user generated content. In: Proceedings of the 2nd Workshop on Analytics for Noisy Unstructured Text Data. 2008, 23–30

24. Blei D, Ng A, Jordan M. Latent dirichlet allocation. Journal of Machine Learning Research, 2003, 3: 993–1022

25. Shen W, Wang J Y, Luo P, Wang M. LINDEN: linking named entities with knowledge base via semantic knowledge. In: Proceedings of the 21st World Wide Web Conference. 2012, 449–458

26. Lao N, Cohen W. Relational retrieval using a combination of path-constrained random walks. Machine Learning, 2010, 81(1): 53–67

27. Tran G, Alrifai M, Nguyen D. Predicting relevant news events for timeline summaries. In: Proceedings of the 22nd International World Wide Web Conference (Companion Volume). 2013, 91–92

28. Tran G, Alrifai M, Herder E. Timeline summarization from relevant headlines. In: Proceedings of the 37th European Conference on Information Retrieval Research. 2015, 245–256

29. Zaragoza H, Rode H, Mika P, Atserias J, Ciaramita M, Attardi G. Ranking very many typed entities on wikipedia. In: Proceedings of the 16th ACM Conference on Information and Knowledge Management. 2007, 1015–1018

Chengyu Wang is a PhD candidate in School of Computer Science and Software Engineering, East China Normal University (ECNU), China. He received his BE degree in software engineering from ECNU in 2015. His research interests include Web data mining, information extraction, and natural language processing. He is working on the construction and application of large-scale knowledge graphs.

Guomin Zhou is an associate professor and vice director at Department of Computer and Information Technology, Zhejiang Police College, China. His research interests include intelligent information processing and big data informatization.

Xiaofeng He is a professor in computer science at School of Computer Science and Software Engineering, East China Normal University, China. He obtained his PhD degree from Pennsylvania State University, USA. His research interests include machine learning, data mining, and information retrieval. Prior to joining ECNU, he worked at Microsoft, Yahoo Labs and Lawrence Berkeley National Laboratory.

Aoying Zhou is a professor in computer science at East China Normal University (ECNU), China where he is heading the School of Data Science and Engineering. He got his master and bachelor degrees in computer science from Sichuan University, China in 1988 and 1985 respectively, and won his PhD degree from Fudan University, China in 1993. Before joining ECNU in 2008, he worked for Fudan University at the Computer Science Department from 1993 to 2007, where he served as the department chair from 1999 to 2002. He worked as a visiting scholar under the Berkeley Scholar Program in UC Berkeley, USA in 2005. He is the winner of the National Science Fund for Distinguished Young Scholars supported by NSFC and the professorship appointment under Changjiang Scholars Program of Ministry of Education. He is now acting as the vice-director of ACM SIGMOD China and Technology Committee on Database of China Computer Federation. He is serving as member of the editorial boards of some prestigious academic journals, such as VLDB Journal, WWW Journal. His research interests include Web data management, data management for data-intensive computing, and in-memory data analytics.