Collective Entity Linking in Web Text: A Graph-Based Method

Xianpei Han
Institute of Software
Chinese Academy of Sciences
Beijing, China
xianpei@nfs.iscas.ac.cn

Le Sun
Institute of Software
Chinese Academy of Sciences
Beijing, China
sunle@nfs.iscas.ac.cn

Jun Zhao
National Laboratory of Pattern Recognition, Institute of Automation
Beijing, China
jzhao@nlpr.ia.ac.cn

ABSTRACT

Entity Linking (*EL*) is the task of linking name mentions in Web text with their referent entities in a knowledge base. Traditional EL methods usually link name mentions in a document by assuming them to be independent. However, there is often additional *interdependence* between different EL decisions, i.e., the entities in the same document should be semantically related to each other. In these cases, *Collective Entity Linking*, in which the name mentions in the same document are linked jointly by exploiting the interdependence between them, can improve the entity linking accuracy.

This paper proposes a graph-based collective EL method, which can model and exploit the *global* interdependence between different EL decisions. Specifically, we first propose a graph-based representation, called *Referent Graph*, which can model the global interdependence between different EL decisions. Then we propose a collective inference algorithm, which can jointly infer the referent entities of all name mentions by exploiting the interdependence captured in Referent Graph. The key benefit of our method comes from: 1) The *global* interdependence model of EL decisions; 2) The *purely collective* nature of the inference algorithm, in which evidence for related EL decisions can be reinforced into high-probability decisions. Experimental results show that our method can achieve significant performance improvement over the traditional EL methods.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information storage and retrieval – *Information Search and Retrieval*.

General Terms

Algorithms, Experimentation.

Keywords

Collective Entity Linking, Collective Entity Disambiguation, Graph-based Entity Linking.

SIGIR'11, July 24–28, 2011, Beijing, China.

1. INTRODUCTION

Recent years have witnessed a clear move from Web of information to Web of knowledge. For example, *Wikipedia*[1] provides a Web collaborative platform for knowledge sharing. The *Read the Web* project[2] is a research effort for the automatic knowledge base population from Web. The intended goal of such efforts is to create knowledge bases that contain rich knowledge about the world's entities, their semantic properties, and the semantic relations between them. One of the most notorious examples is *Wikipedia*: its 2010 English version contains more than 3 millions entities and 20 millions semantic relations (Milne et al. [15]). Such resources have often been used in tasks such as text understanding, word sense disambiguation, etc. They can also be used in IR to help better understand the texts and queries by bridging entity mentions in them with the entities in the knowledge base. There is a clear advantage to do this: it will be possible for a user to identify and explore the background knowledge of the searched item. For example, in Figure 1, by bridging the mentions in a web text with their referent entities *Michael Jordan*, *Chicago Bulls* and *Space Jam*, we can find and explore the related information about these entities in knowledge base, such as its textual descriptions, their entity types and the semantic relations between them(e.g., *Employer-of* and *Actor-of*).

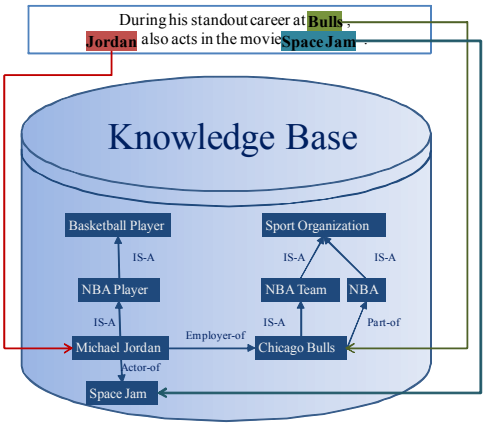

Figure 1. An illustration of entity linking

The key issue is to correctly link the name mentions in a document with their referent entities in the knowledge base, which is usually referred to as *Entity Linking* (*EL* for short). For example, in Figure 1 an entity linking system should link the

[1] http://www.wikipedia.org
[2] http://rtw.ml.cmu.edu/

name mentions "*Bulls*", "*Jordan*" and "*Space Jam*" to their corresponding referent entities *Chicago Bulls*, *Michael Jordan* and *Space Jam* in the knowledge base.

The entity linking, however, is not a trivial task due to the name ambiguity problem, i.e., a name may refer to different entities in different contexts. For instance, the name *Michael Jordan* can refer to more than 20 entities in Wikipedia, some of them are shown below:

- *Michael Jordan(NBA Player)*
- *Michael I. Jordan(Berkeley Professor)*
- *Michael B. Jordan(American Actor)*

To deal with this problem, conventional entity linking approaches have focused on making independent EL decisions using the local mention-to-entity compatibility ([1][3][4][5][6] [12][17][22][24]). The essential idea was to extract the discriminative features (e.g., the important words) from the description of a specific entity (e.g., the Wikipedia page about the entity), then link each name mention in a document by comparing the contextual similarity with each of its candidate referent entities. For example, the name mention "*Jordan*" in Figure 1 will be linked by comparing its contextual similarities with the entities *Michael Jordan*, *Michael I. Jordan* and *Michael B. Jordan*. As these approaches only exploit local features around each name mention, we call these approaches as *local compatibility-based approaches*.

The main drawback of the *local compatibility-based approaches* stems from the fact that they do not take into consideration the *interdependence* between different EL decisions. Specifically, the entities in a topical coherent document usually are semantically related to each other ([11]). For example, in Figure 1, the three entities contained in the document, the NBA player *Michael Jordan*, the NBA team *Chicago Bulls* and the Movie *Space Jam* are all related to each other. In such a case, figuring out the referent entity of one name mention may in turn give us useful information to link the other name mentions in the same document. For example, knowing the mention "*Bulls*" refers to the NBA team *Chicago Bulls* could help us link the mention "*Jordan*" to the basketball player *Michael Jordan* since only this candidate referent entity is semantically related to the *Chicago Bulls*. Similarly, knowing "*Jordan*" refers to the NBA player *Michael Jordan* could in turn help us figuring out the mention "*Bulls*" referring to the NBA team *Chicago Bulls*. These examples strongly suggest that the entity linking performance could be improved by resolving the entity linking problems in the same document jointly, rather than independently. We refer to this approach as *collective entity linking*.

Given a document, the key problem of collective entity linking is to correctly model and exploit the interdependence between the different EL decisions within it. Recent research work has proposed to model the interdependence in a *pair-wise* fashion (Medelyan et al. [16]; Milne & Witten [14]; Kulkarni et al. [11]). These methods model the interdependence between different name mentions as the sum of their *pair-wise* dependencies, and usually determine the referent entity of a name mention by comparing each of its candidate referents with other name mention's referent entities. For example, in Figure 1, the compatibility between the mention "*Jordan*" and the NBA player *Michael Jordan* is the average relatedness between (*Michael Jordan, Space Jam*) and between (*Michael Jordan, Chicago Bulls*). By leveraging the pair-wise dependency, these methods usually can make more accurate EL decisions than the local compatibility based methods, e.g., it can infer that the "*Jordan*" in this example refers to the NBA player *Michael Jordan* due to its relations with *Space Jam* and with *Chicago Bulls*.

However, these methods have a number of limitations: First, the *pair-wise* interdependence model cannot exploit the *global* interdependence between EL decisions, i.e., the structural properties of this interdependence. For instance, in Figure 1, the pair-wise dependence model cannot take into account the implicit dependency between the mention "*Bulls*" and the mention "*Space Jam*", since there is no direct relation between their referent entities *Chicago Bulls* and *Space Jam*. Second, by modeling the interdependence in the pair-wise fashion, the number of computation grows exponentially and the inference process is *NP-hard,* makes the pair-wise model too time-consuming to the real-world applications. Recent pair-wise model based methods ([11][14][16]) usually resolved the inference problem using approximate algorithms, which mostly cannot make the purely collective inference.

To overcome deficiencies of the traditional methods, this paper proposes a graph-based collective entity linking method, which can effectively and efficiently model and exploit the *global* (rather than the *pair-wise*) interdependence between different EL decisions. Specifically, we first propose a graph-based representation, called *Referent Graph*, which can model the global interdependence between different EL decisions as its graph structure. Then we propose a purely collective inference algorithm, which can jointly infer the referent entities of all name mentions in the same document by exploiting both the global interdependence between different EL decisions and the local mention-to-entity compatibility. We have evaluated our method on a standard EL dataset. The experimental results show that our method can achieve significant performance improvement over the traditional EL methods.

The main contributions of this paper are as follows:

1) We propose to model the *global* interdependence between different EL decisions, rather than the *pair-wise* interdependence between them. We also propose the effective *Referent Graph* representation, which can capture the global interdependence as its graph structure;
2) Based on the Referent Graph, we propose a *purely collective* EL algorithm, in which the EL evidence for related name mentions can be collectively reinforced into high-probability EL decisions. These allow us to achieve better performance than the traditional EL methods.

This paper is organized as follows. We first formulate the entity linking problem and briefly review the related work in Section 2. Section 3 describes how to capture the global interdependence between different EL decisions using the Referent Graph. Section 4 describes our method how to exploit the interdependence for collective entity linking. The experimental results are presented and discussed in Section 5. Finally, we conclude this paper and point to some future work in Section 6.

2. THE ENTITY LINKING PROBLEM AND RELATED WORK

In this section, we first formulate the Entity Linking (EL) problem, then compare and contrast the existing EL methods.

ENTITY LINKING (EL) PROBLEM: *Let* $\boldsymbol{M = \{m_1, m_2, \ldots, m_k\}}$ *denote a collection of name mentions. Each name mention* $\boldsymbol{m}$ *in M is characterized by its name* $\boldsymbol{m.S}$*, its local surrounding context* $\boldsymbol{m.C}$ *and the document containing it* $\boldsymbol{m.D}$*. Given a knowledge base* $\boldsymbol{KB}$ *containing a set of entities* $\boldsymbol{\{e_1, e_2, \ldots, e_n\}}$*, the objective of EL is to determine the referent entities in KB of the name mentions in M. Specifically, we use* $\boldsymbol{m.E}$ *to denote the referent entity of a mention m.*

As an example, consider the following EL problem – this will be our running example throughout the paper.

Example 1

NAME MENTIONS: $\{m_1 = Bulls, m_2 = Jordan, m_3 = Space\ Jam\}$

DOCUMENT: *During his standout career at Bulls, Jordan also acts in the movie Space Jam.*

KNOWLEDGE BASE: *Wikipedia*

Here, an EL system should identify the referent entities of the name mentions as $m_1.E$="*Chicago Bulls*", $m_2.E$="*Michael Jordan*" and $m_3.E$ = "*Space Jam*".

In recent years, much research work has focused on the EL problem. Depending on how they model and exploit the interdependence between EL decisions, the existing EL work can be classified into the following three broad categories:

• ***Local Compatibility Based Approaches***: Initial approaches to EL focused on the use of local compatibility based on some context features. As we stated earlier, the idea is to extract the discriminative features of an entity from its textual description, then link a name mention to the entity which has the highest contextual similarity with it. Mihalcea & Csomai [17] proposed a *Bag of Words* (*BoW*) based method, the compatibility between a name mention and an entity was the cosine similarity between them. Cucerzan [4], Bunescu & Pasca [3], Fader et al. [6] extended the *BoW* model by incorporating more entity knowledge such as its categories. Zhang et al. [22] and Mihalcea & Csomai [17] computed the compatibility using classification algorithms. Zheng et al. [23], Dredze et al. [5] and Zhou et al. [24] employed the learning-to-rank techniques, which can take into account the relative rank between the candidate entities. The main drawback of the local compatibility based approaches is that they do not take into account the interdependence between EL decisions.

• ***Simple Relational Approaches***: Observed that EL decisions in the same document are interdependent, Medelyan et al. [16] and Milne & Witten [14] have proposed to compute the mention-to-entity compatibility by leveraging the interdependence between EL decisions. The idea was that the referent entity of a name mention should be coherent with its unambiguous contextual entities. Medelyan et al. [16] determined the compatibility using the semantic relatedness between the candidate entity and the contextual entities. For example, in Example 1, using the only contextual entity *Space Jam*, the compatibility between the mention "*Jordan*" and the NBA player *Michael Jordan* is determined by the semantic relatedness between (*Michael Jordan*, *Space Jam*). Milne and Witten [14] extended the method of Medelyan et al. [16] by adopting learning-based techniques to balance the semantic relatedness, the commoness and the context quality. The drawbacks of these approaches are that they didn't make the collective entity linking, can only exploit the pair-wise interdependence between a name mention and its unambiguous contextual entities, which is usually limited in real world documents. For example, the simple relational approaches will still have difficulty to determine the referent entity *Chicago Bulls* for "*Bulls*" since *Chicago Bulls* is not semantically related to the *Space Jam* – the only unambiguous entity in the context.

• ***Pair-Wise Collective Approaches***: One recent approach proposed by Kulkarni et al. [11] can make collective entity linking, but which only model and exploit the *pair-wise* interdependence between EL decisions. Kulkarni et al. [11] proposed to resolve the collective EL as an optimization problem, where the interdependence between EL decisions is modeled as the sum of their *pair-wise* dependencies. Two approximation solutions were also proposed to resolve the *NP-hard* problem of their inference process.

From the above description, we can see that the interdependence between EL decisions can provide critical evidence for accurate EL decisions. However, all above approaches do not exploit the global interdependence. In the following sections, we demonstrate how the global interdependence can be modeled and exploited using our graph-based collective EL method.

3. THE REFERENT GRAPH

In this section, we propose a graph-based representation, *Referent Graph*, which can capture both the local mention-to-entity compatibility and the global interdependence structure between different EL decisions. In what follows, we first introduce how to model the mention-to-entity compatibility and the interdependence between two EL decisions in the *Referent Graph*, then define the *Referent Graph* in detail.

3.1 Local Mention-to-Entity Compatibility

The local mention-to-entity compatibility measures the likelihood of a name mention m referring to a specific entity e, based on its local context $m.C$. In this paper, the local context of m is the text window around m, and the window size is set to 50 according to the experiments in Pedersen et al. [18].

Traditionally, the compatibility between a name mention m and a specific entity e is determined by the term co-occurrences between the local context of m and the text description of e. Based on the same idea, in our *Referent Graph* we model the local compatibility as a *Compatible* relation between name mention and entity, and the strength of the *Compatible* relation (CP) is calculated based on the *Bag of Words* model:

$$CP(m,e)=\frac{m\cdot e}{|m||e|}$$

where the name mention m is represented as a vector of its context words, and the entity e is represented as a vector of its Wikipedia page's words. All words are weighted using the TFIDF schema.

3.2 Semantic Relation between Entities

As described in Section 1, the interdependence between two EL decisions in a document means that their referent entities should be semantically related to each other. Based on this observation, the *Referent Graph* models the dependency between two EL decisions as a *Semantic-Related* relation between their referent entities. We need a way to measure the strength of the semantic relation between entities, i.e., the semantic relatedness.

There has been several research which focused on computing the semantic relatedness between entities (Strube and Ponzetto [19]; Milne and Witten [9]). In this paper, we adopt the method proposed by Milne and Witten [9] to compute the semantic relatedness between entities, which computes the semantic relatedness as:

$$SR(a,b) = 1 - \frac{\log(\max(|A|,|B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|,|B|))}$$

where a and b are the two entities of interest, A and B are the sets of all entities that link to a and b in Wikipedia respectively, and W is the entire Wikipedia. We show an example of semantic relatedness between four selected entities in Table 1, where the semantic relatedness can reveal the semantic relations between *Michael Jordan* and *Space Jam*, and between *Michael Jordan* and *Chicago Bulls*.

	Space Jam	Chicago Bulls
Michael Jordan	*0.66*	*0.82*
Michael B. Jordan	*0.00*	*0.00*

Table 1. The relatedness table of four selected entities

3.3 The Referent Graph

Now we have two relations: the *Compatible* relation between name mention and entity and the *Semantic-Related* relation between entities. In this way, the interdependence between the EL decisions in a document can be best represented as a graph, which we refer to as *Referent Graph*. Concretely, the *Referent Graph* is defined as follows:

> *A **Referent Graph** is a weighted graph **G=(V, E)**, where the node set **V** contains all name mentions in a document and all the possible referent entities of these name mentions, with each node representing a name mention or an entity; each edge between a name mention and an entity represents a **Compatible** relation between them; each edge between two entities represents a **Semantic-Related** relation between them.*

For illustration, Figure 2 shows the *Referent Graph* representation of the EL problem in Example 1.

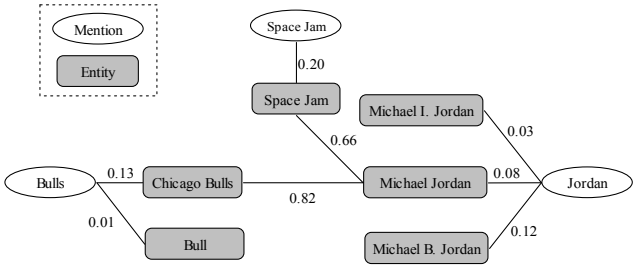

Figure 2. The *Referent Graph* of Example 1

By representing both the local mention-to-entity compatibility and the global entity relation as edges, two types of dependencies are captured in Referent Graph:

1) Local Dependency between name mention and entity.

In Referent Graph, the local dependency between a name mention and an entity is encoded as the edge between the nodes corresponding to them, with an edge weight indicates the strength of this dependency. For example, in Figure 2 the dependency between (*Bulls*, *Chicago Bulls*) is represented as an edge between them. Notice here the dependency between name mention and entity is *asymmetric*: only the entity depends on the name mention, but the name mention does not depend on the entity.

2) Global Interdependence between EL decisions.

By connecting candidate referent entities using the *Semantic-Related* relation, the interdependence between EL decisions is encoded as the graph structure of the Referent Graph. In this way, the referent graph allows us to deduce and use indirect and implicit dependencies, and can take the structural properties of the interdependence into consideration. For example, the name mention *Bulls* is related to the entity *Chicago Bulls*, which in turn is related successively to the entity *Michael Jordan*. An indirect relation between *Bulls* and *Michael Jordan* could be established and used in the EL when necessary. Such indirect relations cannot be identified using an approach based on pair-wise dependence modeling.

The Referent Graph Construction. Given a document, the construction of Referent Graph takes three steps: name mention identification, candidate entity selection and node connection. In this paper, we focus on the task of linking entities with Wikipedia, even though the proposed method can be applied to other knowledge bases. Thus we will only show how to construct the Referent Graph using Wikipedia:

1) **Name Mention Identification**. We first identify all name mentions in a document. Given a document, we gather all N-grams (up to 8 words) and match them to the anchor dictionary of Wikipedia (Medelyan et al. [16]). Not all matches are considered, because even stop words such as "*is*" and "*an*" may match to the anchor texts. We use Mihalcea and Csomai [2]'s *keyphraseness* feature to filter out the meaningless name mentions, and the retained name mentions will be represented in the graph.

2) **Candidate Entity Selection**. In this step, we select the candidate referent entities for each name mention detected in Step 1. We adopt the method described in Milne & Witten [14], where a name mention's candidate referent entities are the destination Wikipedia articles of the anchor text which are the same as this name mention.

3) Node Connection. In this step, we add the dependency edge to the Referent Graph. For each name mention, we add a *Compatible* edge between it and each of its candidate referent entities using the method in Section 3.1. For each pairs of entities in Referent Graph, if there is a *Semantic-Relation* between them, we add an edge between them using the method described in Section 3.2.

4. COLLECTIVE ENTITY LINKING

In this section, we propose a purely collective inference algorithm, which can jointly identify the referent entities of all name mentions in the same document. Given a Referent Graph representation, the goal of the collective EL is to use both the *Compatible* and the *Semantic-Related* relations simultaneously in EL decision making.

4.1 Problem Reformulation

As shown in Section 1, the referent entity of a name mention m should be both:

- Compatible with the name mention m;

- Topically coherent to the other referent entities in the same document.

With respect to the Referent Graph representation, the referent entity of a name mention *m* should be an entity node which has:

- A strong *Compatible* relation with the node corresponding to the name mention *m*;
- Many strong *Semantic-Related* relations with the nodes corresponding to the other referent entities.

To see how this can work, observe in Figure 2 that if we known the referent entities of "*Bulls*" and "*Space Jam*" is *Chicago Bulls* and *Space Jam*, then the *Semantic-Related* relation between (*Chicago Bulls*, *Michael Jordan*) and between (*Space Jam, Michael Jordan*) can provide more evidence for the entity *Michael Jordan* to be the referent entity of "*Jordan*". In contrast, the entity *Michael B. Jordan* lacks the *Semantic-Related* relations with *Chicago Bulls* and *Space Jam*, which suggests that it is not likely to be the referent entity of "*Jordan*", even if it has a stronger *Compatible* relation with the mention "*Jordan*".

But it seems that our method faces a 'chicken-and-egg' problem: The referent entity of a name mention depends on the other referent entities in the same document, and in turn the other referent entities depend on the referent entity of this name mention itself. So how we can resolve it?

In this paper, we resolve such a problem by making a purely collective inference:

1) Firstly, we collect the initial evidence for an entity to be the referent entity of a name mention;
2) Secondly, the evidence is simultaneously reinforced by propagating them between related EL decisions using the interdependence structure in Referent Graph;
3) Finally, our method makes the EL decisions that have the highest probability based on the reinforced EL evidence.

In the following sections, we describe these steps in detail.

4.2 Initial Evidence

Given a document, the initial evidence we used to decide whether an entity is a referent entity of this document is the name mentions in this document and the local compatibilities between these name mentions and their candidate referent entities. Notice that the local compatibility has been encoded as *Compatible* relation in Referent Graph, therefore we only use the name mentions in a document $M = \{m_1, m_2, ..., m_k\}$ as the initial evidence.

The Importance of Evidence. We observed that not all name mentions play equally important roles in the EL process. The referent entities of more important name mentions should provide more information than the referent entities of less important name mentions. For example, consider the following document:

"*The Hall of Fame opens its doors to Michael Jordan, the NBA's greatest player—Yahoo! News*"

The evidence of the name mention *Yahoo! News* should be less important than the evidence of the name mentions *The Hall of Fame*, *Michael Jordan* and *NBA*.

In order to measure the importance of the initial evidence, we assign a *prior* importance score to each name mention. Traditionally, the relative importance of a name mention (notice a name mention is also a phrase) in a document is determined by its TFIDF score. We do the same here: we assign a *prior* importance score to a name mention *m* according to its TFIDF score, and which is further normalized by the sum of TFIDF scores of all name mentions in the same document *D*:

$$\text{Importance}(m)=\frac{tfidf(m)}{\sum_{m\in D} tfidf(m)}$$

Using the above method, we can compute the prior importance of the three name mentions in Figure 2 as Importance(*Bulls*)=0.30, Importance(*Jordan*)=0.25 and Importance(*Space Jam*)=0.45.

4.3 Evidence Propagation

Given the initial evidence, the second step is to reinforce the evidence by propagating them according to the dependency structure captured in Referent Graph. Evidence can be propagated through the two types of edges in Referent Graph in the following way:

Compatible **Edge**: By connecting a name mention and its candidate referent entities, the *Compatible* edge provides a way to propagate evidence from a name mention (i.e., the initial evidence) to its candidate referent entities. Intuitively, a name mention will only propagate evidence to the entity which has a *Compatible* link with it (i.e., will only propagate to its candidate referent entities), and will propagate more evidence to the entity which has a stronger *Compatible* link with it (i.e., has a larger local compatibility *CP(m, e)* between them). Based on the above assumption, we define the evidence propagation ratio from a name mention *m* to an entity *e* as:

$$P(m\rightarrow e)=\frac{CP(m,e)}{\sum_{e\in N_m} CP(m,e)}$$

where N_m is the set of the neighbor entities of a name mention *m*. Notice that there is no evidence propagation from entity to name mention, as *Compatible* edges only go from a name mention to entities. Using this method, we can compute the evidence propagation ratio in Figure 2 as: *P(Space Jam* → Space Jam*)*=1.0, *P(Jordan*→Michael Jordan*)*=0.35 and *P(Jordan*→Michael B. Jordan*)*=0.52.

Semantic-Related **Edge:** By connecting semantically related entities, the *Semantic-Related* edge provides a way to propagate evidence between two related EL decisions by propagating evidence between their referent entities. Intuitively, an entity will only propagate evidence to its neighbor entities, and will propagate more evidence to the entity which has stronger *Semantic-Related* relation with it. Based on the above assumption, we define the evidence propagation ratio from entity e_i to entity e_j as:

$$P(e_i\rightarrow e_j)=\frac{SR(e_i,e_j)}{\sum_{e\in N_{e_i}} SR(e_i,e)}$$

where N_e is the set of neighbor entities of an entity *e*. Using this method, we can compute the evidence propagation ratio between entities in Figure 2, e.g., *P*(Space Jam→Michael Jordan)=1.0 and *P*(Michael Jordan→Space Jam)=0.446.

4.4 Collective Inference

This section describes the collective inference algorithm of our EL method. Specifically, given a name mention *m* in a document *d*, we identify its referent entity as:

$$m.E = \arg\max_{e} CP(m,e) \times r_d(e)$$

where *CP(m,e)* is the local compatibility between *m* and *e*, $r_d(e)$ is the evidence score for entity *e* to be a referent entity of the document *d*. In this way, the EL decision can combine the evidence from the name mention (i.e., the *CP(m,e)*) and the evidence from related EL decisions (i.e., the $r_d(e)$). Since *CP(m,e)* is known in the Referent Graph, the only problem is to compute $r_d(e)$. Here we demonstrate how to jointly compute the $r_d(e)$ for all candidate referent entities of a document *d*, therefore collectively infer the referent entities of all name mentions in a document *d*.

To simplify the description of our algorithm, given a Referent Graph ***G=(V, E)*** contains ***n*** nodes (with ***k*** name mention nodes and ***l*** entity nodes) we assign each node an integer index from 1 to |***V***| and use this index to represent the node, then we can write the adjacency matrix of the Referent Graph ***G*** as ***A***, where ***A[i,j]*** or $\boldsymbol{A_{ij}}$ is the edge weight between node *i* and node *j*. We introduce the follows three notations for our collective EL algorithm:

s: The initial evidence vector, an n × 1 vector where s_i=Importance(*m*) if *i* corresponds to a name mention *m*;

r: The evidence vector, an n×1 vector where r_i is the evidence score for the node *i* to be an referent entity in document *d* (i.e., $r_d(e)$ if node *i* correspond to the entity *e*) or the evidence score contained in this node *i* (if node *i* corresponds to a name mention).

T: The evidence propagation matrix, an n×n matrix where T_{ij} is the evidence propagation ratio from node *j* to node *i*. T_{ij} is computed using the method described in Section 4.3.

In this way, the initial evidence is encoded in the initial evidence vector ***s*** and the interdependence between EL decisions is encoded in the evidence propagation matrix ***T***, the problem is how to get the evidence vector ***r***.

To compute the evidence vector ***r***, we first set its initial value $\boldsymbol{r^0}$ as the initial evidence vector ***s***, i.e.,

$$r^0 = s$$

Then we can update the evidence vector by propagating them according to the interdependence between EL decisions, i.e., the evidence propagation matrix *T*. In this way, we can write the recursive form of the evidence vector as:

$$r^{t+1} = T \times r^t$$

where the $\boldsymbol{r^t}$ is the evidence vector we know at time *t*. One problem of the above formula is that some nodes in the Referent Graph without evidence outward edges, when the evidence propagates to this node it disappears. For example, in Figure 2 the entity node "*Michael I. Jordan*" cannot propagate any evidence to other nodes. To resolve this problem, we introduce a reallocate condition: at each step we reallocate a fraction of evidence to the initial evidence vector ***s***, then we can get the final recursive form of the evidence vector as:

$$r^{t+1} = (1-\lambda) \times T \times r^t + \lambda \times s$$

where $\lambda \in (0,1)$ is the fraction of the reallocation evidence at each step, which is set to the value *0.1* through a learning process in Section 5. By solving this equation, we can get the closed-form solution of evidence vector as:

$$r = \lambda (I - cT)^{-1} s$$

where $c = 1 - \lambda$ and *I* is the identity matrix. In this way, we can jointly infer the $r_d(e)$ value for each candidate referent entity in a document *d*. Observe the above formula, we can see that our collective inference algorithm can combine evidences from the interdependence between EL decisions (***T***), the local compatibility between name mention and entities (***T***) and the relative importance of name mentions (***s***).

Using the above method, we can compute the evidence score for the entities in Figure 2 and the results are shown in Table 2. From Table 2 we can see that: ① our collective inference algorithm can effectively identify the referent entities of a document: the three referent entities *Space Jam*, *Chicago Bulls* and *Michael Jordan* all received a significantly higher evidence score than other candidate entities such as *Bull* and *Michael B. Jordan*; ② The interdependence between EL decisions is critical for the correct EL decision: although the compatibility between (*Jordan*, *Michael Jordan*) is lower than the compatibility between (*Jordan*, *Michael B. Jordan*), the interdependence information can still give the entity *Michael Jordan* a significantly higher evidence score.

Entity	**Space Jam**	**Chicago Bulls**	**Michael Jordan**
$r_d(e)$	*0.144*	*0.202*	*0.314*
Entity	**Bull**	**Michael I. Jordan**	**Michael B. Jordan**
$r_d(e)$	*0.0015*	*0.0045*	*0.018*

Table 2. The evidence scores for entities in Example 1

The Random Graph Walk Explanation of Our Algorithm. Notice that ***s*** is a normalized vector and *T* is a column normalized matrix, we can view the evidence propagation process as a random walk process in graph (Gbel & Jagers [8]), where ***s*** is the starting vector, ***T*** is the random transition probability matrix and ***r*** is the stationary probability of the nodes in Referent Graph. In this perspective, the collective inference algorithm of our method is the same as the *Random Walk with Restart* algorithm (Tong et al. [21]) in graph or the *Personalized PageRank* algorithm (Haveliwala [20]) in IR.

From the random graph walk perspective, we can explain $r_d(e)$ as the probability with which people who read the document *d* would be interested in the entity *e*. For example, using the $r_d(e)$ value, we may predict that 10% people who reading this paper will be interested in the research area *Entity Linking*, and only 0.01% people who will be interested in the NBA player *Michael Jordan*. Therefore, we can also view the entity linking task as the process of predicting which entities people will be interested by reading a document according to both the name mentions and the semantic relations between entities.

The *Posterior* Importance of Name Mention. For many applications such as the Wikification of Web page (Mihalcea & Csomai [17]), only the important name mentions in a document will be linked. Therefore, we need to assign an importance score to the name mentions in a document. In Section 4.2, we have assigned a *prior* importance score to name mention according to its TFIDF score. However, since we have known the referent

entity of a name mention, we can update its *prior* importance score to its *posterior* importance score as follows:

$$\text{Importance}_{\text{post}}(m)=\text{Importance}(m)\times r_d(m.E)$$

5. Experiments

In this section, we assess the performance of our method and compare it with the traditional methods. As most of the earlier work, this paper evaluates the EL method on the task of *linking with Wikipedia*, even though the proposed method can be easily applied to other knowledge bases. In following, we first explain the experimental settings in Section 5.1, then evaluate and discuss the results in Section 5.2.

5.1 Experimental Settings

5.1.1 Knowledge Base

In our experiments, we use the Jan. 30, 2010 English version of Wikipedia[3] as the knowledge base. We prepared the Wikipedia data according to the method described in Hu et al. [25]. In total, the knowledge base we used contains:

- *Over 3,000,000 distinct entities;*
- *A name-to-entity dictionary which contains over 10,000,000 distinct entity names and the candidate referent entities of each name;*
- *Over 20,000,000 semantic relations between entities, which are used to compute the semantic relatedness.*

5.1.2 Data Set

Traditional methods usually used the Wikipedia articles as the ground truth entity linking results ([3][4][14][17]). However, as observed in Kulkarni et al.[11], Wikipedia articles are unsuitable to the evaluation of high-recall entity linking tasks because it annotates name mentions very sparsely (only the important name mentions are annotated). The recent Knowledge Base Population track in TAC 2009[4] (McNamee & Dang [12]) has provided a standard data set for the evaluation of entity linking task, but this data set focuses on individual EL tasks in different documents, and is unsuitable for our collective EL settings.

Due to the above reasons, we adopted the publicly available **IITB** data set[5] to evaluate the performance of our EL method, which is also the data set used in Kulkarni et al.[11]. The IITB data set contains a set of documents (107 documents in total) collected from the web sites belonging to a handful of domains. For each document, its name mentions' referent entities in Wikipedia are manually annotated to be as exhaustive as possible ([11]). In total, 17,200 name mentions are annotated, 161 name mentions per document on average. In our experiments, we evaluate the EL performance using only the name mentions whose referent entities are contained in Wikipedia.

5.1.3 Evaluation Criteria

This paper adopted the same performance metrics used in the Kulkarni et al.[11], which includes ***Recall***, ***Precision*** and ***F1***. Let $\boldsymbol{M}^*$ be the golden standard set of the linked name mentions, $\boldsymbol{M}$ be the set of linked name mentions outputed by an EL method, then these metrics are computed as:

$$Precision = \frac{|M \cap M^*|}{|M|}$$

$$Recall = \frac{|M \cap M^*|}{|M^*|}$$

Notice here that two name mentions are considered equal if and only if their names *m.S*, the documents containing them *m.D* and their referent entities *m.E* are all equal (Notice there is no guarantee that a method can achieve the *Recall* of 1.0). In the same way as in Kulkarni et al.[11], *Precision* and *Recall* are averaged across documents and overall *F1* is computed from average *Precision* and *Recall*. Because *Precision* and *Recall* are often negatively correlated, they do not always get their peaks at the same time. In this case, we used *F1* as the primary performance metric.

5.1.4 Baselines

We compare our method with four state-of-the-art baselines:

Wikify!. This is the same EL method described in Mihalcea & Csomai [17], which is a standard *Local Compatibility* based method. The *Wikify!* computes the local compatibility using the contextual overlap between the name mention and the dictionary definitions of the entity(in their method, its Wikipedia page).

Cucerzan. This is the same method described in Cucerzan [4]. This is also a *compatibility* based method, but the compatibility between a name mention *m* and a candidate referent entity *e* is determined by two factors: the standard local compatibility and the relatedness between *e* and all other name mentions' candidate referent entities in the same document.

M&W. This is the same EL method described in Milne & Witten [14], which is a state-of-the-art *simple relational method*. Given a name mention, *M&W* determines its referent entity by (mainly) comparing each of its candidate referent entity using the average relatedness between the entity and the name mention's unambiguous contextual entities.

CSAW. This is the same EL system described in Kulkarni et al.[11]. As described in Section 2, the *CSAW* is a collective EL method based on the *pair-wise* EL decision interdependence modeling.

Except for the *CSAW*, all other three baselines are designed only to link the important name mentions (i.e., key phrases) in a document. In our experiment, in order to compare the performances in a high recall, we push these systems' recalls by reducing their importance thresholds of linked name mentions.

5.2 Experimental Results

5.2.1 Overall Performance

We conduct experiments on the **IITB** data set with several methods: the baselines *Wikify!*, *Cucerzan*, *M&W* and *CSAW*, our method using only the local compatibility (i.e., the referent entity of a name mention is simply the entity which has the largest compatibility with this name mention)—which we denote as ***Our Method(LC)*** and the full model of our method—***Our Method***. For all methods, the parameters were configured through 10-fold

[3] It can be obtained from http://download.wikipedia.org for free research use

[4] http://projects.ldc.upenn.edu/kbp/data/

[5] http://www.cse.iitb.ac.in/~soumen/doc/QCQ/

cross validation. The overall performance results are shown in Table 3.

	Precision	Recall	F1
Wikify!	*0.55*	*0.28*	*0.37*
Cucerzan	*0.71*	*0.33*	*0.45*
M&W	*0.80*	*0.38*	*0.52*
CSAW	*0.65*	*0.73*	*0.69*
Our Method(LC)	***0.52***	***0.34***	***0.41***
Our Method	***0.69***	***0.76***	***0.73***

Table 3. The overall results on IITB data set

From the results in Table 3, we can make the following observations:

1) By modeling and exploiting the global interdependence between different EL tasks, our collective EL method can achieve significant performance improvements over the traditional methods: compared with the local compatibility based baseline *Wikify!*, our method can produce a 36% F1 improvement; compared with the simple relational baseline *M&W*, our method can produce a 21% F1 improvement; compared with the pair-wise based collective baseline *CSAW*, our method can produce a 4% F1 improvement.

2) The interdependence between the referent entities in the same document can provide critical evidence to the EL decision: ① By adding the relatedness between entities into the local compatibility, the *Cucerzan* can achieve a 8% F1 improvement over the local context based baseline *Wikify!*; ② By using only the relatedness between entities, the simple relational method *M&W* can achieve a 15% F1 improvement over the local context based baseline *Wikify!*.

3) By modeling and exploiting the interdependence between different EL tasks, the collective EL method can achieve significant performance improvement over the independent decision making based EL methods: ① Compared with the local compatiblity based baseline *Wikify!*, the two collective EL methods *CSAW* and *Our Method* can significantly improve the F1 measure by 32% and 36% respectively; ② Compared with the simple relational baseline *M&W*, the two collective EL methods *CSAW* and *Our Method* can significantly improve the F1 measure by 24% and 28% respectively; ③ By exploiting the interdependence and making the collective inference, *Our Method* can achieve a 32% F1 improvement over our local compatibility only system—*Our Method(LC)*.

4) By modeling and exploiting the global interdependence, our method can further improve the EL performance than the *pair-wise* interdependence model: compared with the pair-wise interdependence model based *CSAW* baseline, our method can achieve a 4% F1 improvement. We believe this is because our Referent Graph can encode more interdependence between EL tasks than the pair-wise based model, and our purely collective inferent algorithm can better exploit the global interdependence structure between different EL decisions, in particular, regarding indirect relations and mutual reinforcement.

5.2.2 Disambiguation Precision

For many EL applications such as the Wikification of Web pages and Wikipedia articles, the disambiguation precision plays a critical role. However, because precision and recall are strongly related, it is not straightforward to compare the disambiguation precision of different EL systems. Observed that in our experiments all EL systems can achieve a recall larger than 20%, we compare the disambiguation precision of different EL methods at the recall of 20%. The results are shown in Table 4.

	Precision
Wikify!	*0.60*
Cucerzan	*0.71*
M&W	*0.83*
CSAW	*0.87*
Our Method(LC)	***0.58***
Our Method	***0.87***

Table 4. The Precision results @20% Recall

From Table 4, we can make the following observations:

1) The local context typically is not enough to support a high disambiguation precision: even at the recall of 20%, the two local context based methods, *Wikify!* and *Our Method(LC)*, can only achieve precisions of 0.60 and 0.58.

2) The interdependence between referent entities is the better evidence for disambiguation than the local context of name mentions: all the three interdependence based methods (*M&W*, *CSAW* and *Our method*) can achieve a precision higher than 0.80 at the recall of 20%.

5.2.3 Recall-Precision Tradeoff

Usually, the disambiguation precision of an EL system is related to the recall of name mentions. To discuss the tradeoff between recall and precision, we show the precisions at different recalls of our method in Figure 3 (Notice that the highest *Recall* that can be achieved by *Our Method* and *Our Method(LC)* are correspondingly *0.83* and *0.32*). Furthermore, in order to intuitively demonstrate the EL result when pushing the recall into a higher value, Table 5 shows our method's results of the top 20% weighted and the last 20% weighted name mentions of the Wikipedia article *Michael I. Jordan*[6].

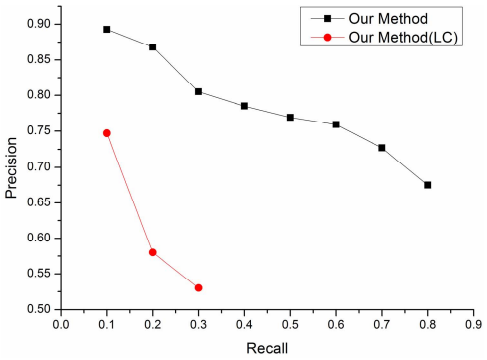

Figure 3. The Precisions vs. Recalls

[6] http://en.wikipedia.org/wiki/Michael_I._Jordan

	Name Mention	Referent Entity	Score
Top 20%	*Michael I. Jordan*	Michael I. Jordan	*0.0038*
	machine learning	Machine learning	*0.0033*
	formalisation	Formal system	*0.0029*
	cognitive	Cognitive science	*0.0025*
	statistics	Statistics	*0.0024*
	variational methods	Variational Bayesian methods	*0.0023*
	PhD	Doctor of Philosophy	*0.0022*
	Neural Networks	Neural network	*0.0019*
	EM	Expectation-maximization algorithm	*0.0019*
	artificial intelligence	Artificial intelligence	*0.0019*
	NSF	National Science Foundation	*0.0018*
	UC, Berkeley	University of California, Berkeley	*0.0014*
Last 20%	*paper*	Academic publishing	*7.92E-4*
	full professor	Professor	*7.29E-4*
	David E. Rumelhart	David Rumelhart	*6.47E-4*
	Zoubin Ghahramani	Zoubin Ghahramani	*6.25E-4*
	Presidential	President	*6.10E-4*
	application	Computer software	*5.99E-4*
	Pioneer Award	GLAAD Media Awards	*5.50E-4*
	R	R (programming language)	*4.89E-4*
	ACC	Atlantic Coast Conference	*4.50E-4*
	Investigator	Player character	*4.28E-4*
	Lawrence	D. H. Lawrence	*4.19E-4*
	American Control Conference	American Automatic Control Council	*3.67E-4*

Table 5. The EL results of the Wikipedia article of Berkeley professor *Michael I. Jordan*

From Figure 3 and Table 5, we can see that:

1) Our method can achieve high EL precision on the **important name mentions** of a document. Figure 3 showed that our method can achieve 0.90 precision at the recall of 0.1, and 0.87 precision at the recall of 0.2 (As shown in Section 4, when we push the recall of our method, we add the name mentions according to their relative importance in a document).

   We believe this is because important name mentions are more coherent to the topic of a document than less important name mentions, therefore the document contains more evidence for the EL decisions of these name mentions. For instance, in Table 5, the top 20% name mentions are all topical coherent to the topic of this document, i.e., the professor *Michael I. Jordan*, such as his research areas *machine learning* and *statistics*, his school *UC Berkeley*, etc. In contrast, the last 20% are less topical coherent to the topic, such as the name mentions *Presidential*, *application* and *R*.

2) Compared with the local compatibility based methods, the collective EL methods shows a greater advantage in linking less important name mentions in a document. Figure 3 shows that when we push the recall from 0.1 to 0.3, the performance improvement of Our Method increases from 15% to 18%. We believe this is because the EL method can capture more evidence than the local compatibility based methods: except for the local surrounding context, but also the interdependence between different EL decisions.

3) For real world usage, we must take the trade-off between precision and recall into consideration. For some applications like the identification of missing links in Wikipedia (Adafre & Rijke [1]) and the wikification of Web page (Mihalcea & Csomai [17]), the precision is critical, therefore we need to loose the recall to achieve a higher precision. For instance, we may only link the top 20% name mentions in Table 5 to hold a high entity linking precision. In contrast, for some applications where recall is critical like the topic indexing in IR (Medelyan et al. [16]) and web people search (Artiles et al. [2]), we need to loose the precision for a high recall.

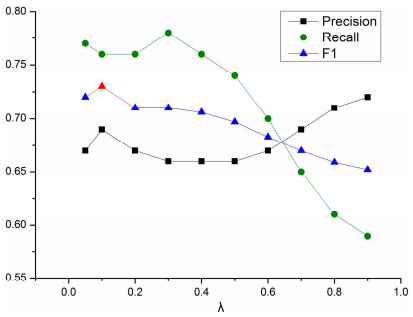

Figure 4. The Performance results vs. λ

5.2.4 *Optimizing Parameters*

Our method has only one parameter λ, which is the reallocation fraction of the evidence in every evidence propagation step. Intuitively, a larger λ will increase the importance of the initial evidence and the local compatibility in the collective EL decision; and at the same time decrease the importance of the interdependence between EL tasks. Figure 4 plots this trade-off. As shown in Figure 4, our method can achieve the best F1 performance when the value of λ is 0.1.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a graph-based method to the entity linking task. This method can collectively infer the referent entities of all name mentions in the same document. By modeling and exploiting the global interdependence between different EL decisions, the proposed method can achieve competitive performance over the traditional methods.

In our method, we did not take into account the *NIL entity problem* of the EL task, i.e., the referent entity of a name mention may not be contained in the given knowledge base. For future work, we will resolve this aspect in our graph-based method by leading a pseudo NIL entity into our model. Furthermore, using the entity linking method, we want to develop a Web entity search and mining system by annotating billions of Web pages with their entity information.

7. ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China under Grants no. 60773027, 60736044, 90920010, 61070106, 60875041 and 61003117, and the National High Technology Development 863 Program of China under Grants no. 2008AA01Z145. Moreover, we sincerely thank the reviewers for their valuable comments.

8. REFERENCES

[1] Adafre, S. F. & de Rijke, M. 2005. *Discovering missing links in Wikipedia*. In: Proceedings of the 3rd international workshop on Link discovery.

[2] Artiles, J., Sekine, S. & Gonzalo, J. 2008. *Web people search*. In: Proceedings of LREC, vol. 8.

[3] Bunescu, R. & Pasca, M. 2006. *Using encyclopedic knowledge for named entity disambiguation*. In: Proceedings of EACL, vol. 6.

[4] Cucerzan, S. 2007. *Large-scale named entity disambiguation based on Wikipedia data*. In: Proceedings of EMNLP-CoNLL.

[5] Dredze, M., McNamee, P., Rao, D., Gerber, A. & Finin, T. 2010. *Entity Disambiguation for Knowledge Base Population*. In: Proceedings of COLING.

[6] Fader, A., Soderland, S., Etzioni, O. & Center, T. 2009. *Scaling Wikipedia-based named entity disambiguation to arbitrary web text*. In: Proceedings of Wiki-AI at IJCAI.

[7] Gabrilovich, E. and Markovich, S. 2007. *Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis*. In: Proceedings of the IJCAI.

[8] Gbel, F. & Jagers, A. A. 1974. *Random walks on graphs*. In: Stochastic processes and their applications, vol. 2, no. 4, pp. 311-336.

[9] Han, X. & Zhao, J. 2009.*Named Entity Disambiguation by leveraging Wikipedia semantic knowledge*. In: Proceedings of CIKM.

[10] Han, X. & Zhao, J. 2010. *Structural semantic relatedness: a knowledge-based method to named entity disambiguation*. In: Proceedings of the 49th ACL.

[11] Kulkarni, S., Singh, A., Ramakrishnan, G. & Chakrabarti, S. 2009. *Collective annotation of Wikipedia entities in web text*. In: Proceedings of the 15th ACM SIGKDD.

[12] Li, X., Morie, P. & Roth, D. 2004. *Identification and tracing of ambiguous names: Discriminative and generative approaches*. In: Proceedings of AAAI, pp. 419-424.

[13] McNamee, P. & Dang, H. T. 2009. *Overview of the TAC 2009 Knowledge Base Population Track*. In: Proceeding of Text Analysis Conference.

[14] Milne, D. & Witten, I. H. 2008. *Learning to link with Wikipedia*. In: Proceedings of the 17th ACM CIKM.

[15] Milne, D., et al. 2006. *Mining Domain-Specific Thesauri from Wikipedia: A case study*. In: Proceedings of WI.

[16] Medelyan, O., Witten, I. H. & Milne, D. 2008. *Topic indexing with Wikipedia*. In: Proceedings of the AAAI WikiAI workshop.

[17] Mihalcea, R. & Csomai, A. 2007. *Wikify!: linking documents to encyclopedic knowledge*. In: Proceedings of the sixteenth ACM CIKM.

[18] Pedersen, T., Purandare, A. & Kulkarni, A. 2005. *Name discrimination by clustering similar contexts*. In: Proceedings of CICLing.

[19] Strube, M. and Ponzetto, S. P. 2006. *WikiRelate! Computing Semantic Relatedness Using Wikipedia*. In: Proceedings of AAAI.

[20] Taher H. Haveliwala. 2003. *Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search*. IEEE Transactions on Knowledge and Data Engineering.

[21] Tong, H., Faloutsos, C. & Pan, J. Y. 2007. *Fast random walk with restart and its applications*, *Data Mining*. In: Proceedings of ICDM.

[22] Zhang, W., Su, J., Tan, Chew Lim & Wang, W. T. 2010. *Entity Linking Leveraging Automatically Generated Annotation*. In: Proceedings of the 23rd COLING.

[23] Zheng, Z., Li, F., Huang, M. & Zhu, X. 2010. *Learning to Link Entities with Knowledge Base*. In: The Proceedings of NAACL.

[24] Zhou, Y., Nie, L., Rouhani-Kalleh, O., Vasile, F. & Gaffney, S. 2010. *Resolving Surface Forms to Wikipedia Topics*. In: Proceedings of the 23rd COLING.

[25] Hu, J., Fang, L., Cao, Y., et al. 2008. *Enhancing Text Clustering by Leveraging Wikipedia Semantics*. In Proceedings of SIGIR.