Collective Entity Linking on Relational Graph Model with Mentions

Jing Gong[1[00]] ,Chong Feng [1[00]] ,Yong Liu[2[11]] , Ge Shi[1[22]] and Heyan Huang[1[00]]

[1] Beijing Institute of Technology University, Beijing 100081, China
[2] State Key Lab, , Beijing 100081, China
{gongjing,fengchong,hhy63}@bit.edu.cn [11]24787806@qq.com
[22]shige713@126.com

Abstract. Given a source document with extracted mentions, entity linking calls for map-ping the mention to an entity in reference knowledge base. Previous entity linking approaches mainly focus on generic statistic features to link mentions independently. However, additional interdependence among mentions in the same document achieved from relational analysis can improve the accuracy. This paper propose a collective entity linking model which effectively leverages the global interdependence among mentions in the same source document. The model unifies semantic relations and co-reference relations into relational inference for se-mantic information extraction. Graph based linking algorithm is utilized to ensure per mention with only one candidate entity. Experiments on datasets show the proposed model significantly out-performs the state-of-the-art relatedness approaches in term of accuracy.

Keywords: Collective Entity Linking, Entity Disambiguation, Relational Graph.

1 Introduction

The Entity Linking (EL) is crucial for information extraction and knowledge base population [1-3]. Given a document and a list of extracted mentions such as people, locations, organizations, entity linking targets at mapping the mention to an entity from reference knowledge base (KB) like Wikipedia, DBpedia, or YAGO etc. For example, considering the sentence posted to a news story: “Browne and Caldwell talked about the ongoing security crackdown in Baghdad”. The mentions “*Browne*” and “*Caldwell*” should be mapped to the entities “*Sam Browne*” and “*Reche Caldwell*” respectively. The mentions are ambiguous because most of people are “*Browne*” and “*Caldwell*”.

Most of earlier EL approaches focused on generic statistical features, which were later enhanced with a certain level of global reasoning. But essentially most approaches fail to acquire and exploit context semantic information in source documents. For example, “*He is supporting Gordon Brown, David Cameron is also backing Brown*”, the mention “*Brown*” should be mapped to “*Gordon Brown*”. If we notice that the latter “*Brown*” refers to the first Gordon “*Brown*”, it will be much easier to link the latter mention to the corresponding entity instead of linking it to the dominant “*Brown*” (like most existing entity link systems did). The main idea of this intuition is to understand that the relationship between the two "*Brown*" is a pair of co-reference. Similarly, other relationship between the mentions will also provide us efficient semantic features for the linking task.

To address the shortcomings of features-based method, it is intuitive to consider the analysis of context semantic information and make fully use of relations be-tween mentions and entities. An example of our method is shown in Figure 1, which illustrates the map between mentions and entities from reference knowledge base. Through analyzing the context, the mentions in the same document are semantically related to each other. We also exploit the relationship among entities in the knowledge base, which stores a huge amount of explicit information. To the end, we utilize a graph based algorithm in term of these relations to disambiguate the entities. Our collective EL methods jointly exploit the interdependence between mentions in the same document, while non-collective approaches linking each mention independently.

This paper is organized as follows. Section 2 describes related work. Section 3 discuss how to exploit the semantic relations and co-reference relations in source documents. The construction of relational graph and a graph based linking algorithm are presented to disambiguate entities in Section 4. The experimental results are given and discussed in Section 5.

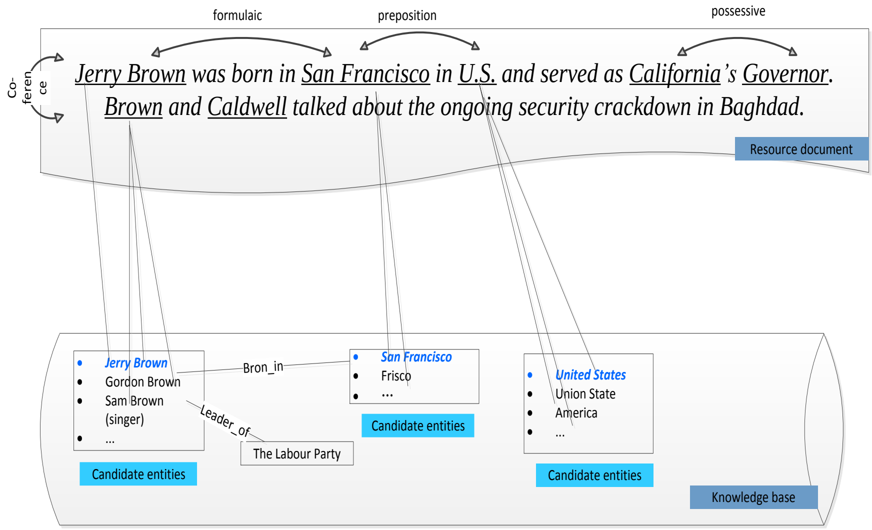

Fig. 1. The example of Entity Linking.

2 Related Work

Early works on entity linking often formulated the task as a Word Sense Disambiguation (WSD) problem [8-9], which determined the correct sense of a multi-meaning word in a text according to its context information. The EL approaches can be divided into the following three major categories:

1) Local and Global Compatibility based Approaches: The *local* compatibility method focused on context discriminative features to map a mention to the entity which has the highest contextual similarity. Fader et al. [5] defined a similarity measure that compared the context of a mention to the Wikipedia categories of an entity candidate. The *global* compatibility method focused on all mentions in a document simultaneously to arrive at a coherent set of disambiguation and utilized the link structure information to estimate coherence. Ratinov et al. [4], Cucerzan et al. [6], proposed to emphasize different coherence measures between the titles of the disambiguated mentions in the same document and the relatedness of common noun phrases in a mention's context. Milne&Witten et al. [7] leveraged semantic relatedness between a mention's candidate entities and the unambiguous mentions in the textual context. While these features pointed towards semantic coherence and were still limited to mapping each mention separately. Even though these local and global compatibility based approaches had a competitive coverage rate, it would not work well on highly ambiguous surface strings.
2) Relational based Approaches: The *relational* based approaches focused on computing the relationship of candidate entity-to-candidate entity and mention-to-candidate entity. Their motivation utilized the coherent and interdependent mentions in the same document. Dutta et al. [8] proposed a joint model combining cross-document co-reference resolution and entity linking, which also focused on co-occurring mentions allowing for global context and feature propagation. Zheng et al. [9] applied a dynamically joint inference method to improve within-document co-reference resolution. However, these approaches did not exploit the global interdependence among mentions in the same document and suffered on high computational costs even if for an approximate solving of the optimization model.
3) Graph based Approaches: Navigli and Lapata et al. [10] proposed the graph connectivity metrics method, in which nodes were ranked with respect to their local importance of centrality measures such as in-degree, centrality, PageRank or HITS, etc. Blanco et al. [11] made a connection between graph problem and the Maximum Capacity Representative Set. Aharonu et al. [12] leveraged queries, websites and Wikipedia ideas collaboratively for getting to know generic search space intents and assemble a heterogeneous graph to characterize a number of kinds of relationships between them. Han et al. [13] and Liu et al. [14] proposed the graph-based collective entity linking algorithm, which utilized structured relationship of the knowledge base and external knowledge sources. While these approaches ignore the semantic information between mentions in the same document, which could improve the entity linking accuracy.

3 Relation Extraction

The primary challenge in incorporating relational analysis into the entity linking task is to systematically construct the relational constraints. We explore semantic relations and co-reference relations for relational inference. Not only the textual relations are extracted from the text, but also the weights are assigned to these semantic relations. Different from previous work on relation extraction [21], which are mainly conducted on ACE2004 (Automatic Content Extraction) or Relation Detection and Characterization

(RDC) dataset, our method utilize large scale knowledge resources effectively, such as Wikipedia and YAGO.

3.1 Derivation of the Semantic Relations

The relation types in knowledge base are categorized into 4 semantic relations [15], which are {*premodifiers*, *possessive*, *preposition*, *formulaic*}. More detailed description of the four structures are as followings:

1) Premodifiers: modifies the proper adjective or proper noun. E.g.: [the [Chinese] building]

2) Possessive: indicates the first mention. E.g.: [[California]'s Governor]

3) Preposition: indicates two semantically related mentions by the existence of a preposition. E.g.: [[The Great Wall] in [China]]

4) Formulaic: indicates formulaic relations according to the ACE04 annotation guideline. E.g.: [The Great Wall], [China].

Algorithm 1 Exploiting Semantic Relations

Input: $\boldsymbol{M} = \{\boldsymbol{m_1}, \boldsymbol{m_2} \dots \boldsymbol{m_k}\}$is the set of mention. **S**= {premodifiers, possessive, preposition, formulaic}.M_{train} is the set of annotated gold mentions in training data. $\boldsymbol{D_g} = \{(\boldsymbol{m_i}, \boldsymbol{m_j}\} \in \boldsymbol{M_{train}} \times \boldsymbol{M_{train}} \boldsymbol{D_s} = \emptyset$

Output: cluster $\mathbf{R} = \emptyset$ of syntactic relations.

RE_{base} = RE classifier trained on D_g;
foreach $(\boldsymbol{m_i}, \boldsymbol{m_j}) \in \boldsymbol{M}$ **do:**
  p= structure inference on $(\boldsymbol{m_i}, \boldsymbol{m_j})$using patterns;
if $\mathbf{p} \in \mathbf{S} \vee (\boldsymbol{m_i}, \boldsymbol{m_j})$ was annotated with a S structure **do:**

$$\boldsymbol{D_s} = \boldsymbol{D_s} \cup (\boldsymbol{m_i}, \boldsymbol{m_j})$$

RE_s = RE classifier trained on D_s;

for each $(\boldsymbol{m_i}, \boldsymbol{m_j}) \in \boldsymbol{M}$ **do:**
  p = structure inference on $(\boldsymbol{m_i}, \boldsymbol{m_j})$ using patterns;
  if p∈ S **do:**
    r = relation for $(\boldsymbol{m_i}, \boldsymbol{m_j})$ using RE_s;
    R = R∪ r;
  else do:
    r = relation for $(\boldsymbol{m_i}, \boldsymbol{m_j})$ using RE_{base};
    R = R∪ r;
return R

Fig. 2. The algorithm of exploiting syntactic relations

The process of semantic relations extraction is illustrated in algorithm 1. Before employing the algorithm of semantic relations extraction, a pre-processing phase is necessary to improve the accuracy. The processing of mentions is aided by mention expansion and segmentation. Since some entities may have different names, aliases, acronyms and abbreviations, we use regular expressions to match abbreviations and longer surface forms that are often incorrectly segmented or ignored by NER due to different annotation standards.

3.2 Derivation of the Semantic Relations

Understanding of co-reference relations are also important for entity linking. Considering the following example:

"Jerry Brown was born in San Francisco in U.S. and served as California's Governor, Brown and Caldwell talked about the ongoing security crackdown in Baghdad.".

The mention "*Brown*" should be mapped to the entity "*Jerry Brown*", but all existing EL approaches would map the popular page of *Brown*.

Thus, besides semantic relations, the co-reference relations are encountered to cover the common cases, where two or more co-reference mentions are mapped to the same entity. In this process, the input is the set of candidate entities mapped by mentions, and the output is the cluster C of co-reference relations. The entities, that share tokens or be acronyms of others, are clustered in the following algorithm 2.

```
Algorithm 2 Exploiting the Co-reference Relations

Input: M = {m_1, m_2 ... m_k} is the set of mention. θ is the cutoff
threshold.
Output: co-reference cluster C.
        For all m_i, m_j ∈ M do:
            If C[m_i] ≠ C[m_j] and Sim(C[m_i], C[m_j]) > θ do:
            Merge(C[m_i], C[m_j], C);
        end for
        return C
```

Fig. 3. The algorithm of exploiting the co-reference relations

An issue occurs that the correct co-referent candidate entity might not exist in the candidate list in the cluster. To resolve the problem, we ignore candidate entities generated from short surface strings and give it the same candidate list as the head mentions in its cluster. The processing of longer and shorter mentions are different because the shorter mentions are inherently more ambiguous. The longer mentions should collectively refer to shorter mentions once a co-referent relation is determined.

3.3 Global Optimization of Relations by Integer Linear Programming

Our objective function of relational inference can be defined as following:

$$\Gamma = \sum_i \sum_k P(t_i^k|m_i)\, e_i^k + \sum_i \sum_k Z\alpha sim(\sigma|m_i) r_{ij}^{(k,l)} \quad (1)$$

Where

- m_i: the *i*-th mention.
- $\boldsymbol{t_i^k}$.: the *k*-th candidate title being chosen for mention m_i.
- $P(t_i^k|m_i)$: the initial score for the *k*-th candidate title being chosen for mention m_i.

$e_i^k \in \{0,1\}(\forall i \sum_k e_i^k = 1)$ is used to denote whether we disambiguate $\boldsymbol{m_i}$ to t_i^k.

The relation is denoted as $r_{ij}^{(k,l)} \in \{0,1\}(r_{ij}^{(k,l)} = e_i^k \wedge e_j^l, 2r_{ij}^{(k,l)} \leq e_i^k + e_j^l)$ whether title t_i^k and t_j^l are chosen simultaneously. Its value depends on the textual relation type and on how coherent it is with our existing knowledge.

Z is a normalization factor that normalizes all $\sum_i \sum_k Z\alpha sim(\sigma|m_i)$ to the range [0, 1]. The symbol α is the weight of implicit relations with explicit predicate, whose range is [1,5].

$\sigma = (t_i, \mathrm{p}, t_j)$ is the set of triples obtained from indexing all Wikipedia links and DBpedia relations. The arguments t_i, t_j are tokenized, stemmed and lowercased, *p* is a relation predicate from the DBpedia ontology or the predicate linking indicating a hyperlink relation.

The integer linear programming problem is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers. The objective function and the constraints (other than the integer constraints) are linear.

4 Entity Disambiguation on Relational Graph

4.1 Construction of Relational Graph

A weighted, undirected relational graph G= <V, E> is constructed, where $\mathrm{V} = \{v_1, v_2 \ldots v_m\}$ is the set of nodes, namely mentions and candidate entities, and $\mathrm{E} = \{e_1, e_2 \ldots e_n\}$ is the set of edges. The goal of this relational graph is to identify a dense sub-graph that contains merely one mention-entity edge for each mention.

Mention-Mention Graph

To avoid abusing linguistic knowledge from the source documents, we construct a collective mention-mention graph, whose edges are the selected semantic relations and co-reference relations. Figure 4 depicts a constructed mention-mention graph, which contains a set of vertices representing the mentions extracted from the source document and a set of undirected edges. The weights of the edges are calculated by equation (1).

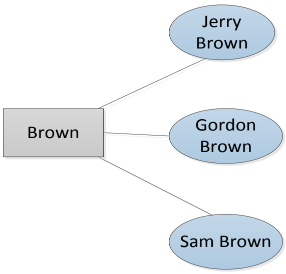

Fig. 4. The example of Mention-Mention Graph.

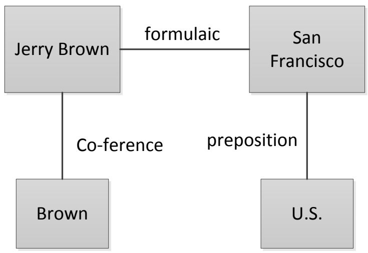

Fig. 5. The example of Mention-Entity Graph.

Mention-Entity Graph

We construct the mention-entity graph with mentions and its candidate entities as vertexes, whose weights of edges are calculated by the context similarity between mentions and its corresponding candidate entities such as $sim(context(m_i), context(t_i^k))$, the $context(m_i)$ denotes the context window around mention m_i, the $context(t_i^k)$denotes the context window around the anchor of candidate entity in the Wikipedia page.

Entity-Entity Graph

We utilize the semantic relations of types and classes between entities in knowledge base to construct the entity-entity graph. The weights of edges are calculated by the equation $P(e_i, e_j) = 1 - \frac{log(max(|IN_{e_i}|,|IN_{e_j}|)) - log(|IN_{e_i} \cap IN_{e_j}|)}{log(|N|) - log(min(|IN_{e_i}|,|IN_{e_j}|))}$, IN_{e_i} denotes the number of incoming links of candidate entity e_i. Figure 6 presents a sub-graph containing the relevant entities in the *Jerry Brown* example.

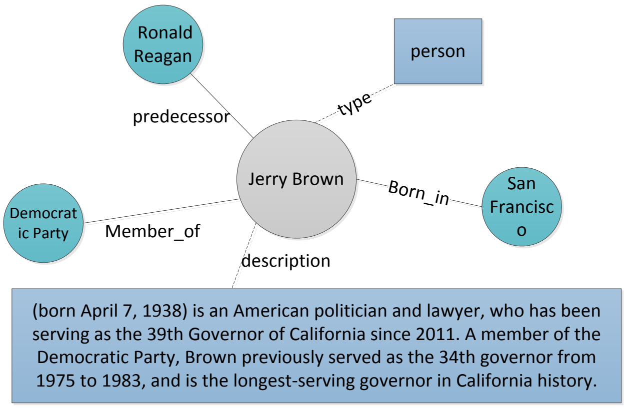

Fig. 6. The example of Entity-Entity Graph

4.2 Graph based Linking Algorithm

The goal of this graph based linking algorithm is to calculate a dense sub-graph which would ideally contain all mention nodes and exactly one mention-entity edge for each mention. The challenge of this task is how to specify a notion of density which is best suited for capturing the coherence of the resulting entity nodes.

Algorithm 3 Graph based Linking Algorithm

Input: the relational graph G, the set of mentions M, the set of candidate entities E.
Output: result graph with one edge per mention.
For each mention **do**
Calculate W_{m_i, m_j} , m_i ***is connected with*** m_j;
for each candidate entity **do:**
if the candidate entity is isolated and the weight is minimum, drop the nodes;
keep the closest(5*mentions_count) candidate entities, drop others;
while each mention has more than one candidate entity **do:**
if m_i ***and*** m_j ***has edge*** & e_i^k ***and*** e_j^l ***has edge*** do
$S = S \cup \{e_i^k, e_j^l\}$;
else
Set the candidate entity with highest weighted degree to S;

Fig. 7. Graph based Linking Algorithm

Hence we need to pay attention to the weak links in the collective entity set of the desired sub-graph. We regard the value of relational inference between nodes in the graph as the total weights of its incident edges. The density of a sub-graph could be regarded as the minimum weighted degree among its nodes. Based on an approximation algorithm proposed by Sozio et al. [16], this paper propose a graph based linking algorithm to find strongly interconnected, size-limited groups in the graph.

Let S be equal to the set of candidate entities per mention in relational graph. The definition of $\boldsymbol{e_i^k}$ is a candidate entity of$\boldsymbol{m_i}$, and let $\boldsymbol{N_{m_i}}$be the number of all mentions. Let $\boldsymbol{N}_{m_i,e_i^k}$be the number of candidate entities of $\boldsymbol{m_i}$.

5 Experiments and Analysis

5.1 Experiments System

The process of entity linking contains three steps which is illustrated in the following. Firstly, we generate initial mentions M= {mi} from source documents and candidate entities C= {ci} from knowledge base, then extract semantic and co-reference relations from source documents with regular expressions. Finally, we leverage the integer linear programming to integrate the semantic information and disambiguate mentions in term of graph based linking algorithm.

5.2 Preparation of Dataset

To evaluate the performance of our method, we conduct experiments on 4 datasets used in Ratinov et al. [4]. The ACE dataset is a subset of ACE2004 Coreference documents and MSNBC is from Cucerzan et al.[6]. The AQUAINT dataset is introduced in Milne&Wittenet al. [7] and the Wiki dataset is a subset of Wikipedia. The detailed statistics are presented in Table 1.

Table 1. The description of 4 datasets

dataset	The number of Text	The number of Linking
ACE	57	620
MSNBC	20	150
AQUAINT	50	449
Wiki	80	700

For each mention, we check whether the KB entity returned by EL approach is correct or not. Standard metrics is adopted in the following to evaluate the experimental performance. Let M* be the golden standard set of the linked mentions, M be the set of linked mentions outputted by EL system. We get Precision (P), Recall (R) and F1 score (F1) through equations as follows:

$$\mathrm{P} = \frac{|M \cap M*|}{|M|}, \mathrm{R} = \frac{|M \cap M*|}{|M*|}, \mathrm{F1} = \frac{2(P+R)}{P*R} \quad (2)$$

5.3 Comparison of different approaches of Entity Linking

We evaluate and compare our results with five approaches, which are *Tf-idf, Wikification* [17], *Aida* [18], *M&W*[7],*R&R*[4], *List-only*[21]. ***LGSCR*** (Entity Linking with Reference Graph Model) represents method proposed by this paper.

1) ***Tf-idf**:* A simple *local compatibility* based method using the context similarity between mentions and candidate entities.
2) ***Wikification**:* This approach identified entity relations and interdependence among mentions. Then incorporates these relations into an integer linear programming formulation.
3) ***Aida**:* This is an integrated EL method which unifies prior probability and text similarity into a weighted *graph model*. The Aida utilizes robustness tests for self-adaptive behavior to avoid some specific situations.
4) ***M&W:*** Milne&Witten utilized *simple relational features* between candidate entities and the context mentions.
5) ***R&R***: Ratinov utilized *local and global features* for entity disambiguation to Wikipedia.
6) ***List-only***: Lin selected seed mentions by *collective inference* to bridge the gap between mentions and non-informative target entities.
7) ***LGSCR**:* Our approach utilize collective inference to link a set of coherent mentions simultaneously, which combines semantic relations with co-reference relations, integrating these relations into a graph based linking algorithm.

As Table 2 demonstrates, our collective EL method significantly outperforms other approaches. The ***LGSCR*** scored 2.41% higher than *aida* system in F1 score, which jointly exploits the global interdependence among mentions for entity disambiguation. By utilizing the semantic relations between mentions and entities, the *Wikification* achieve a higher performance over the generic statistical based baseline *tf-idf*.

Through the investigation of the four systems we find that the statistical method *tf-idf* does not suffice to the specific situations. For example, the sentence "*Instead of Los Angeles International, for example, consider flying into Burbank or John Wayne Airport in Orange County, Calif.*", the mention "*Burbank*" can be mapped to the wrong entity "*Burbank,California*" with high *tf-idf*, however, our EL system map the correct entity "*Bob_Hope_Airport*" according to the semantic relations.

Table 2. Accuracy (%) of different methods on test set

Approaches	ACE	MSNBC	AQUAINT	Wiki	TAC2014
Tf-idf	73.52	72.99	73.75	79.77	73.02
Wikification	84.25	83.83	84.91	89.68	87.11
Aida	85.77	85.10	86.43	88.76	86.23
M&W	82.44	84.06	83.55	87.45	82.09
R&R	83.22	85.03	85.67	90.01	85.28
List-only	84.65	83.83	85.87	88.49	85.92
LGSCR	**85.98**	**87.35**	**86.96**	**92.44**	**87.57**

Moreover, the performance in ***LGSCR*** is higher than other EL approaches, which is probably because they mainly reflect the relations among mentions and not the importance of the word itself just as position and frequency do. Consequently, this experiment not only demonstrates the effectiveness of graph based linking algorithm, but also reveals the importance of global interdependence structure among mentions to entity linking.

5.4 Analysis of Features

In this section, we incrementally add five components to the system and explore their impacts on the linking performance. We chose five groups of features: *local features, global features* [4], *semantic relations, co-reference relations,* and *relational graph*.

1) *Local* features capture the context similarity between the vector of mention context and candidate entity context.
2) *Global* features are refinements of similarity measures among Wikipedia titles, which leverage the incoming or outgoing link structure in Wikipedia. Thus we utilize a well-known Pointwise Mutual Information (PMI) relatedness measure. Given a Wikipedia title collection W, titles t1 and t2 with a set of incoming links L1, and L2 respectively, PMI is computed as follows: $\mathrm{PMI}(\mathrm{L}_1, L_2) = \frac{|\mathrm{L}_1 \cap L_2|/|W|}{|\mathrm{L}_1|/|W||L_2|/|W|}$
3) *Semantic relations* include global interdependence between mentions in the same source document. The algorithm of exploiting semantic relations is presented in section 3.1.
4) *Co-reference relations* include different surface mentions mapped to a same entity. An algorithm of exploiting the co-reference relations is illustrated in section 3.2.
5) *Relational Graph* integrates semantic relations and co-reference relations into graph based linking algorithm. The construction of relational graph and the graph based linking algorithm are given in section 4.

Table 3 shows the performance of our EL system with different features. The final results are highly improved after adding relations among mentions and entities, which is probably because the relational inference can explore the implied semantic information and the interdependence among mentions. Compared with the local and global features, the semantic and co-reference relations of interdependence among mentions can significantly improve the F1 measure by 3.10%. By exploiting the relational graph model, our EL method can further improve the performance by 2.44% than the measure of semantic and co-reference relations.

Error analysis in many cases has shown that the summaries of the different disambiguation candidates for the same surface forms are very similar. The disadvantage of this approach is that irrelevant candidates are inevitably added to the disambiguation context, which would create noises. Different characteristics show somewhat consequently different gains from the various aspects of our approach.

LR : local features + relational graph.

LGR: local features + global features + relational graph.

LGSR : local + global features + semantic relations features + relational graph.

LGSCR : local + global + semantic relations features+ co-reference relations features + relational graph.

Table 3. Results of entity linking with different groups of features (F1 %)

Methods	ACE	MSNBC	AQUAINT	Wiki	TAC2014
LR	81.71	81.12	83.33	87.91	81.21
LGR	83.54	83.02	84.65	88.28	83.32
LGSR	84.59	84.14	85.97	89.86	84.42
LGSCR	**86.73**	**85.44**	**86.78**	**90.14**	**87.36**

6 Conclusions and Future Work

This paper propose a novel collective entity linking method, which jointly exploit the interdependence among mentions by selecting the most coherent set of entity candidates on the KB side. The model effectively incorporates semantic relations and co-reference relations into a graph based linking algorithm. The experiment results reveal that it performs better than all other state-of-art approaches with different features. In the future, more relations such as temporal relations and conjunction relations could be considered for entity linking task.

Acknowledgement

The research of this paper is partially supported by National 863 project 2015AA015404 and open project of State key lab. Smart manufacturing for special vehicles and transmission system.

References

1. Heng Ji, Ralph Grishman, Dang H T, et al. Overview of the TAC 2010 knowledge base population track[C]// Text Analysis Conference. 2010.
2. http://nlp.cs.rpi.edu/kbp/2016/taskspec.pdf.
3. Ji H, Grishman R. 2011. Knowledge Base Population: Successful Approaches and Challenges.[C]// Proceedings of ACL, 19-24 June, 2011, Portland, Oregon, Usa..
4. Ratinov L, Dan R, Downey D, et al. Local and Global Algorithms for Disambiguation to Wikipedia.[C]// Proceedings of ACL, 19-24 June, 2011, Portland, Oregon, Usa. 2011:1375-1384.
5. Fader A, Soderland S, Etzioni O. Scaling Wikipedia-based Named Entity Disambiguation to Arbitrary Web Text [J]. Proc of Wikiai, 2009.
6. Cucerzan S. TAC entity linking by performing full-document entity extraction and disambiguation[C]//Proceedings of the Text Analysis Conference. 2011, 2011.
7. Milne D, Witten I H. Learning to link with wikipedia[C]//Proceedings of the 17th ACM conference on Information and knowledge management. ACM, 2008: 509-518.
8. Dutta S, Weikum G. C3EL: A Joint Model for Cross-Document Co-Reference Resolution and Entity Linking[C]// Proceedings of EMNLP. 2015.

9. Zheng J, Vilnis L, Singh S, et al. Dynamic Knowledge-Base Alignment for Coreference Resolution[C]// Conference. 2013:153-162.
10. Navigli R, Lapata M. An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2010, 32(4):678-92.
11. Blanco R, Boldi P, Marino A. Using graph distances for named-entity linking[J]. Science of Computer Programming, 2015.
12. Aharonu M, Kale M R. Entity Linking based Graph Models for Wikipedia Relationships [J]. International Journal of Engineering Trends & Technology, 2014, 18(8):380-385.
13. Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In SIGIR, pages 765–774. ACM, 2011.
14. Qiao L, Yun Z, Yang L, et al. Graph-Based Collective Chinese Entity Linking Algorithm[J]. Journal of Computer Research & Development, 2016.
15. Chan Y S, Roth D. Exploiting syntactico-semantic structures for relation extraction[C]// Proceedings of ACL: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, Usa. 2011:551-560.
16. Sozio M, Gionis A. The community-search problem and how to plan a successful cocktail party[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010:939-948.
17. Xiao Cheng,and Dan Roth. 2013. Relational Inference for Wikification. [C]// Proceedings of EMNLP, ACL, 2013:1787-1796.
18. Hoffart J, Yosef M A, Bordino I, et al. Robust disambiguation of named entities in text[C]// Proceedings of EMNLP, ACL, 2011:782-792.
19. Flanigan J, Thomson S, Carbonell J, et al. A Discriminative Graph-Based Parser for the Abstract Meaning Representation[C]// Proceedings of ACL. 2014:1426-1436.
20. Ji, H., Nothman, J., & Hachey, B. 2014, November. Overview of tac-kbp2014 entity discovery and linking tasks. In Proc. Text Analysis Conference (TAC2014).
21. Ying Lin, Chin-Yew Lin, Heng Ji. List-only Entity Linking. [C]// Proceedings of ACL. 2017:536-541.