COLLECTIVE LIST-ONLY ENTITY LINKING: A GRAPH BASED APPROACH

Seminar Report

submitted in partial fulfillment of the requirement
for award of Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ Abdul Kalam Technological University

Submitted by

ANIKA BABU

Department of Computer Science and Engineering
Mar Athanasius College of Engineering
Kothamangalam

COLLECTIVE LIST-ONLY ENTITY LINKING: A GRAPH BASED APPROACH

Seminar Report

submitted in partial fulfillment of the requirement
for award of Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

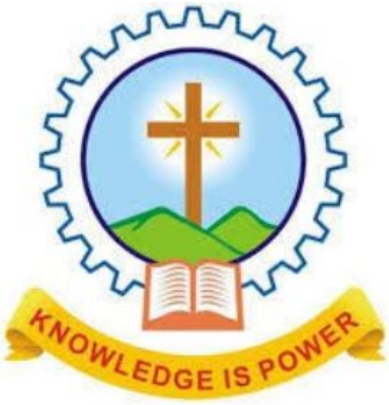
APJ Abdul Kalam Technological University

Submitted by

ANIKA BABU

Department of Computer Science and Engineering
Mar Athanasius College of Engineering
Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM



CERTIFICATE

This is to certify that the report entitled **Collective List-Only Entity Linking: A Graph-Based Approach** *submitted by* : **Ms. ANIKA BABU** *, Reg. No.* **MAC15CS012** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by her under our supervision and guidance.*

.....	.....	.....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head of the Department***

Date: Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the 'God Almighty' for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby George and Prof Joby Any Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

List-only entity linking (EL) is the task of mapping ambiguous mentions in texts to target entities in a group of entity lists. Traditional EL task uses rich semantic relatedness in knowledge bases to improve linking accuracy. List-only EL can merely take advantage of co-occurrences information in entity lists. The current method utilizes co-occurrences information to enrich entity descriptions. The local compatibility between mentions and entities are calculated to determine results. Entity coherence is also deemed to play an important part in EL, which is currently neglected. In addition to local compatibility, this approach takes into account global coherence among entities to harness co-occurrences in entity lists for mining both explicit and implicit entity relations. The relations are then integrated into an entity graph. PageRank is incorporated to compute entity coherence. The final results are derived by combining local mention-entity similarity and global entity coherence. The experimental studies validate the superiority of this method. It not only improves the performance of the list-only EL, but also opens up the bridge between the list- only EL and conventional EL solutions.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

EL	Entity Linking
KB	Knowledge Base
NER	Named Entity Recognition
LSTM	Long Short Term Memory
NLTK	Natural Language Tool Kit

Introduction

Entity Linking (EL) is the task of detecting corresponding named entities for ambiguous mentions in text. Mention refers to character string, such as Jackson in the example shown in Fig. 1, the true meaning of which needs to be determined by being linked to an entity, such as the basketball coach Phil Jackson. Traditional EL methods leverage knowledge bases (KBs), which offer rich semantic information of entities, for robust and accurate disambiguation process. Nevertheless, despite the effectiveness of knowledge-based EL, it might not be applicable in situations where there is insufficient information of entities, such as entity lists.

Entity list, as is often the case, consists of a group of closely-related entities, and it exists in various information sources [1]. In contrast to KBs, where complete structure of entities facilitates almost all entity-related tasks, entity list minimizes necessary information to mere co-occurrences of interrelated entities, thus serving as a light-weight alternative in terms of describing entity correlations.

Entity lists can be found useful, for instance, in the scenario concerning detection of emerging stock names. When investors search new stock names in Wikipedia,1 a frequently updated KB, chances are that there are no corresponding items.In fact as shown in [2] for a dataset including 2,468 stock names, merely 340 of them can be found in Wikipedia. Nevertheless, those stocks can be found co-occurring with others in stock lists on financial websites. Thus, the stock lists will be of great use if people have doubts concerning new stocks. There are much more similar situations, such as searching for specific car brands or collecting information about bars in a small town, where the knowledge about target entities is sparse.

Consequently, the demand for list-only EL emerges [1], which targets at solving the problem of mapping ambiguous mentions to entity lists (rather than KBs); Fig. 1.1 describes an example of list-only EL problem. State-of-the-art method [1] addresses the challenge by merely considering the local compatibilities between mentions and entities to determine matching pairs, whereas neglecting the global coherence among entities.

Traditional Entity Linking (EL) technologies rely on rich structures and properties in the target knowledge base (KB). However, in many applications, the KB may be as simple and sparse as lists of names of the same type (e.g., lists of products). We call it as List-only

Entity Linking problem. Fortunately, some mentions may have more cues for linking, which can be used as seed mentions to bridge other mentions and the uninformative entities. In this work, we select the most linkable mentions as seed mentions and disambiguate other mentions by comparing them with the seed mentions rather than directly with the entities. Linking mentions to automatically mined lists show promising results and demonstrate the effectiveness of this approach.

Several research results have shown that specifying the information about certain entities is the most common information demand of information retrieval users. The needs should be answered by returning specific entities, their properties or related concepts instead of just any type of documents. While some search engines are capable of recognizing specific types of entities, true entity-oriented search still has a long way to go because of the high ambiguity in names across documents. Entity linking (EL) goes beyond the entity recognition task by linking a textual named entity mention to a knowledge base entry. It is a difficult task involving several challenges. It's application can be found in the biomedical domain. In addition, results of the latest EL work are provided for reference, which uncover new EL challenges found in biomedical text mining, along with discussions regarding their possible solutions.

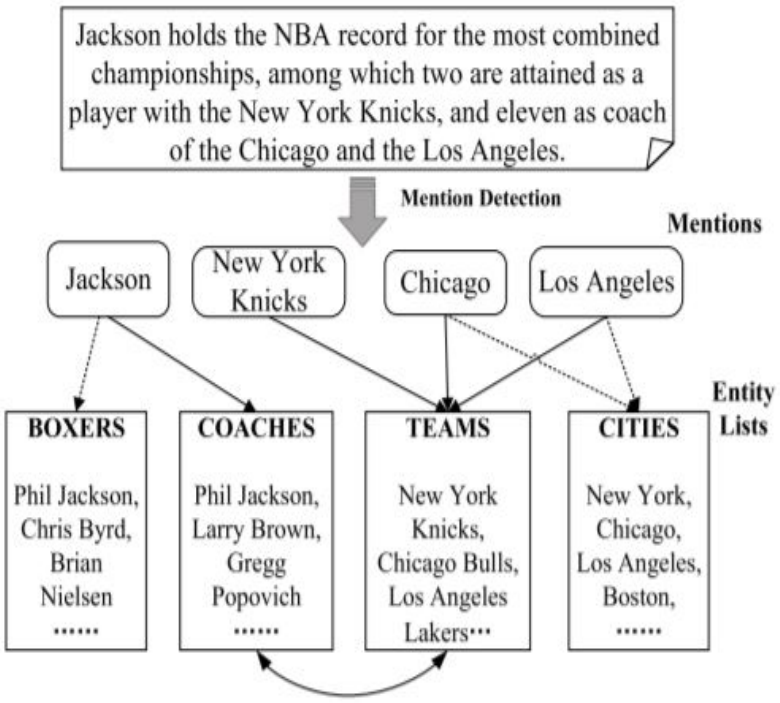


Figure 1.1: Example of list-only entity linking.

Example 1 : As shown in Fig 1.1 there is a piece of text with mentions Jackson, New York Nicks, Chicago and Los Angeles; and there are 4 sample entity lists to be linked to, namely Boxers, Coaches, Teams and Cities. The task of list only EL is to link mentions to correct entities in the entity lists. It can be seen that entity Chicago and entity Los Angeles in entity list Cities have the same name strings with mention Chicago and mention Los Angeles in the text. Because of the high mention-entity compatibility, existing method tends to map mentions Chicago and Los Angeles to entities Chicago and Los Angelesin the entity list featured Cities. However, the true entities for them are Chicago Bulls and Los Angeles Lakersin entity list Teams. Furthermore, it is hard for current method to decide which entity that mention Jackson should be linked to, since there are two possible candidate entities with the same name Phil Jackson and they are in different entity lists.

Moreover, the dataset used for empirical study might be inappropriate and need a re-design. Current dataset is comprised of documents, which contain mentions to be disambiguated, and a group of entity lists, which include the true entities for mentions. However, each document only contains a single mention for disambiguation, which may not reflect the reality well. A pragmatic scenario may look like the example in Fig. 1, where there are four mentions in one document. Additionally, the entities in different entity lists are dissimilar, making the task much easier to cope with since each mention may well only have one candidate entity. This also deviates from reality and simplifies the problem.

In short, the shortcomings of the existing list-only EL solution are two-fold :

• Entity coherence within or across entity lists was overlooked and not leveraged;
• Results were supportless for lack of appropriate dataset and deliberate experiment design.

In particular, we propose to solve list-only EL task by taking account of the correlations in entities and converting the disambiguation problem to a graph problem. We show the merits of graph-based list-only EL by referring to the example in Fig. 1.1 It is easy to map mention New York Knicks to entity New York Knicks in the entity list featured Teams. Then by considering the interdependence of entities in the same list Teams, mention Chicago will be mapped to entity Chicago Bulls, and mention Los Angeles will be mapped to entity Los Angeles Lakers. Additionally, by further taking into account cross-dependence of entities across different entity lists, entity Phil Jackson in the Coaches entity list, rather than entity Phil Jackson in the Boxers entity list, will be chosen as the target entity for mention Jackson.

To implement graph-based list-only EL, we mainly carry through the following three steps.
(1) Pre-processing— including an optional named entity recognition process and the candi-

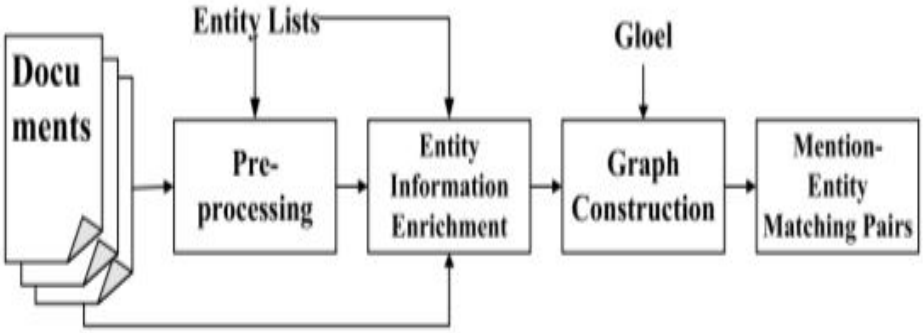


Figure 1.2: Flowchart of graph-based list-only EL.

date entity generation process. This step formalizes raw texts and produces mentions and candidate entities as inputs for later steps.

(2) Entity information enrichment. The descriptions of entities are enriched by collecting representative texts from the inputs, which in turn enable the establishment of coherence among entities.

(3) Graph-based entity disambiguation. An entity graph is constructed by integrating outputs from earlier steps.

We propose a graph-based algorithm **Gloel**, which implements Personalized **Page Rank** to determine how likely an entity is the target entity by taking into consideration both coherences among entities, and compatibilities between mentions and entities. The outputs are a list of pairs comprised of mentions and their most possible entities.

Furthermore, we put forward a new procedure to construct datasets applicable to evaluating list-only EL. The experimental results in this new dataset validate the effectiveness of graph-based linking, a popular method of collective linking, and the in-depth analysis shows that compared with existing list-only linking method, our graph-based solution achieves better performance in list-only EL task.

1.1 Contributions

The main contributions of this article can be summarized into three ingredients:

• We motivate to revise list-only EL by taking into account relations between entity lists, i.e., global coherence, in addition to local compatibilities between mentions and entities.

• We tackle the problem by a graph-based method and offer a new algorithm Gloel, where Personalized PageRank is adopted to capture global coherence among candidate entities.

• A new dataset construction procedure is presented to cater to the redefined task, and Gloel is experimentally evaluated on top of it, and shown to outperform state-ofthe-art method.

Given a piece of text, locate the mentions of relevant entities and refer each mention to a single entity. This task is similar to, but not the same as, deciding where to put explanatory Wikipedia links if you were editing a Wikipedia page.

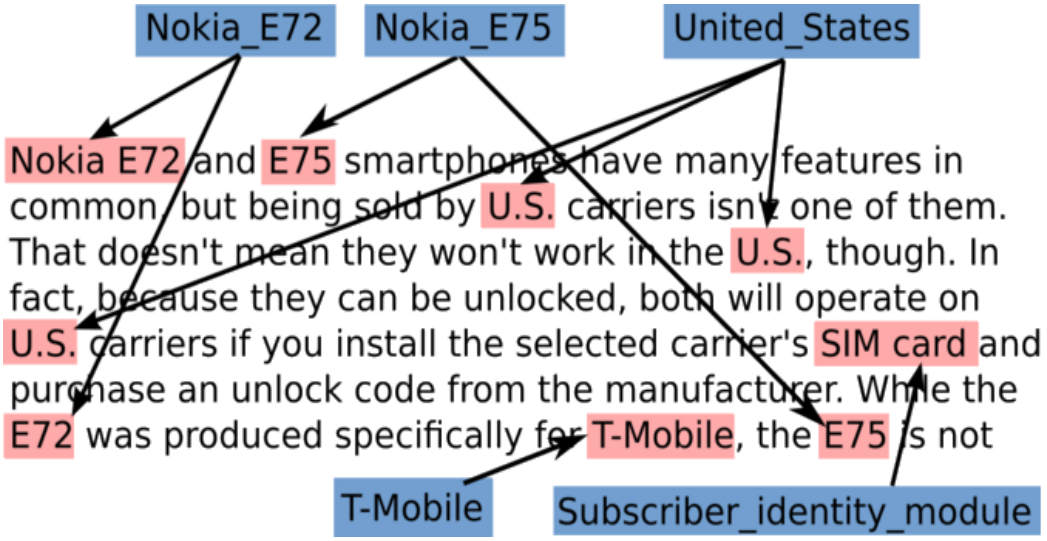


Figure 1.3: example 2 - EL

The figure 1.3 shows an example annotation of mentions and entities (shaded in text) that a human is expecting to receive from the entity-linking system for the given text. Note that common words like smart phone or manufacturer, which do match with entities in Wikipedia, are not expected to be linked to them in this context.

The figure 1.4 lists the entities that our system considers for disambiguation, having detected the mentions and matched them with the knowledge base. These steps must be done carefully, since we do not want to miss any entity, but having too many loosely matching entities would harm the performance of the system. The mention E75 does not suggest the correct entity, however, all of E72, Nokia, and Nokia E72 do include the right candidates. During disambiguation, we compute a score for each candidate entity, and then proceed only

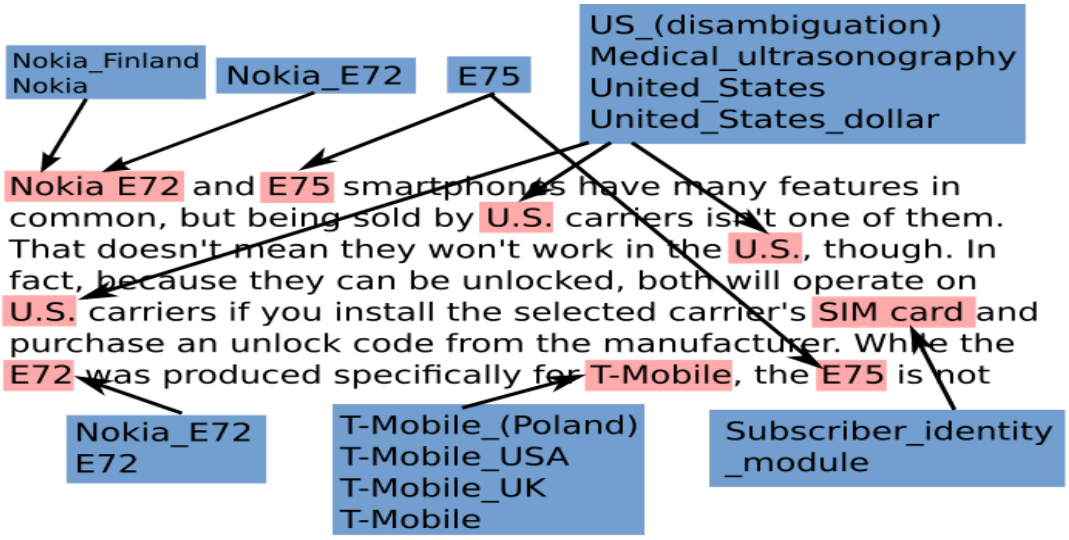


Figure 1.4: Example 2 - EL with all possible entites

with the best ones. We need to handle adjacent and overlapping matches and prune the entities further in order to provide only those found in the entity collection.

1.2 Organization

This report is organized as follows. Chapter 2 summarizes related work and bridges list-only EL with conventional KB-oriented EL.In chapter 3 , the new definition of list-only EL problem and the methodology, which contains three steps, are elaborated. New dataset construction and experiment results and further analysis are detailed in chapter 3 followed by conclusion in Chapter 4

Related work

In this chapter, we brief related work, and discuss the differences and connections between list-only EL and traditional EL.

2.1 List-only entity linking

Over recent years, in accordance with the emergence of various text sources, EL tasks in new forms have been put forward. List-only EL is the task of mapping mentions to a group of entity lists, rather than complete KBs. Entity linking has received much more attention. The purpose of entity linking is to link the mentions in the text to the corresponding entities in the knowledge base. Most work of entity linking is aiming at long texts, such as BBS or blog.It selected seed mentions for each entity list to bridge the gap between mentions and non-informative target entities, and then conducted the independent linking process to determine final results. Noticing that they merely harnessed entity lists co-occurrences information for generating entity descriptions, in this work, we further utilize the co-occurrences information to model entity relatedness and integrate it in the entity graph, which yields a more robust and accurate EL framework when confronting difficult input texts.At present, we simply consider all co-occurring mentions of entities in the same list

Four main challenges can cause numerous difficulties when developing an entity linking system:

i) the kind of textual documents to annotate (such as social media posts, video subtitles or news articles)

ii) the number of types used to categorise an entity (such as Person, Location, Organization, Date or Role)

iii) the knowledge base used to disambiguate the extracted mentions (such as DBpedia, Wikidata or Musicbrainz)

iv) the language used in the documents.

Among these four challenges, being agnostic to the knowledge base and in particular to its coverage, whether it is encyclopedic like DBpedia or domain-specific like Musicbrainz, is arguably the most challenging one. To tackle those four challenges and in order to be knowledge base agnostic, a method that enables to index the data independently of the schema and vocabulary is being used. More precisely, index is designed such that each entity has at least two information: a label and a popularity score such as a prior probability or a Pagerank score. The indexing approach allows to generate an accurate set of candidates from any knowledge base that makes use of linked data, respecting the required information for each entity, in a minimum of time and with a minimal size.

There are other new forms of EL problems which are similar to the list-only task. One is the Target Entity Disambiguation problem . The main disparity is that the focus of Target Entity Disambiguation task lies in finding documents related to the entities given a entity list, whereas the starting point of list-only EL task is to eliminate the ambiguity in documents by using entity lists. Another similar task is the Named Entity Disambiguation with Linkless KBs.Different from the mere entity lists in our task, there are still textual descriptions for entities in Linkless KBs.

2.2 Knowledge base oriented entity linking

The large number of potential applications from bridging web data with knowledge bases have led to an increase in the entity linking research. Entity linking is the task to link entity mentions in text with their corresponding entities in a knowledge base. Potential applications include information extraction, information retrieval, and knowledge base population. However, this task is challenging due to name variations and entity ambiguity.

Earlier work on EL focus on the situation where abundant information exists on the entity side. Specifically, KBs such as YAGO, Freebase and Wikipedia, offer rich semantic structures among entities as well as detailed textual descriptions, thus resulting in robust and accurate linking procedure. KB-oriented EL work can generally be divided into independent and collective methods.

In the former approach, mentions are disambiguated merely according the similarity between mentions and entities, and the problem is transformed into candidate entities ranking so as to obtain the most possible result. The similarity is mainly measured by lexical features such as bagof-words of surrounding texts and statistical features such as prior popularities of

entities. Then as for ranking process, unsupervised methods calculate cosine similarities of feature vectors and output the results, whereas supervised approaches construct classifiers by training on annotated dataset, and the linking process is in the charge of classifiers when inputs are given. Although methods of this kind can achieve good results, semantic coherences within entities are neglected, which prove to be essential in improving overall performances.

With respect to collective linking methods in conventional EL task, most of them assume mentions in the same document are semantically coherent, which also should fit in the textual topic of the whole document. Therefore, the resulting entities also are expected to have high relatedness and the problem is in turn converted to find matching pairs maximizing the coherence. Cucerzan proposed to harness Wikipedia categories to model coherence among entities, while Milne and Witten reckoned normalized Google Distance as another useful tool for measurement to form integer linear programming problem so as to collectively obtain results. Hoffart et al defined keyphrase relatedness to capture entity coherence, and proposed to construct a mention-entity graph, on which dense sub-graph generation algorithm was put forward to determine the subgraph containing one-to-one mention-entity matches. The method of re-formalizing the linking problem by constructing mention-entity or entity-only graph distinguished itself among other works due to its capability to integrate both local similarity information between mentions and entities, along with the coherence information among entities. Based on this, several works [3]–[6] proposed and applied modified graph algorithm on the graph, which improved the disambiguation accuracy and the adaptability to difficult texts. Overall, the collective linking methods generally perform better than the independent counterparts in terms of conventional KB-oriented EL.

2.3 Discussion on differences and connections

There are indeed many similarities between these two lines of works, despite of the evident differences. The disparity mainly lies in the information on the entity side. Regarding conventional KB oriented EL, entities have rich and well structured descriptions offered by KBs, in terms of both text description and internal links among entities. Thereafter, researchers merely need to filter valuable information to improve linking results. In stark contrast, with respect to list-only scenarios, the mere information existing on the entity side is the co-occurrences among entity name strings in the same entity list, which in turn requires information mining and enrichment. In this paper, to avoid help from structured or semi-structured knowledge source, the dataset itself is leveraged to harvest the relevant relations among entities, thus fulfilling the entity information enrichment task.

Nevertheless, aside from information mining process, the methods utilized in conven-

tional research can be applied to this newly-defined problem and will achieve promising results. Above all, the techniques developed in traditional EL also apply in list-only EL problem, and the extra work for the latter is to mine information on the entity side.

The proposed methodology

We start with defining the proposed problem. Existing work defined list-only EL as mapping a single mention mi in document d_i to the corresponding entity $e_{i,j} \in E_j$ in the entity lists. Nonetheless, on the one hand, in most real-life documents, there are more than one mention, differentiating this definition from reality. On the other hand, the ambiguity between entity lists is not stressed, which can turn the problem of mapping mentions to a group of highly ambiguous entities into determining whether the mentions have corresponding entities in the entity lists. And the latter also deviates from the original motivation of EL task, which centres on disambiguating mentions from several possible meanings. We will further elaborate the definition of ambiguity between entity lists via mathematical equations in Chapter 3.

As a consequence, it is vital to extend the definition of this task so as to cater to broader scenarios. Specifically, we formalize list-only EL problem as follows.

Definition 1 [List-Only Entity Linking]: Given a set of documents D = $\{d_1, \ldots d_n\}$, each of which contains a set of mentions $M_i = \{m_{i1}, \ldots m_{is}\}$, an ambiguous set of entity lists $\varepsilon = \{E_1, \ldots E_l\}$, the task is to determine the most possible entity $e_{ij,k} \in E_k$ for each mention m_{ij}, or return NIL if there is no corresponding entity.

Note that the set of entity lists has to be ambiguous to follow the motivation of EL task. In other words, for the majority of entities, there ought to be at least one more ambiguous entity in the entity lists.

We take the example in Fig. 1.1 to explain the definition. There are four mentions to be disambiguated in the document. By utilizing list-only EL, mentions New York Knicks, Chicago, Los Angeles should be mapped to entities New York Knicks, Chicago Bulls, Los Angeles Lakers in the entity list featured Teams respectively, instead of New York, Chicago, Los Angeles in the entity list featured Cities. And mention Jackson should be linked to entity Phil Jackson in the Coaches entity list, rather than entity Phil Jackson in the Boxers entity list.

OVERVIEW : The specific procedure for graph-based list-only entity linking includes three steps, namely, preprocessing, entity information enrichment and graph disambiguation. As shown in Fig. 1.2, the former two steps generate inputs, based on which the entity graph

is constructed and Gloel is performed to determine results.

3.1 Preparation for graph inputs

This subsection presents treatment of raw text data and generation of inputs for graph construction.

3.1.1 Pre-processing

In the pre-processing step, mentions in the text are detected and the candidate entities are also generated. Specifically, the initial input for EL is a set of raw documents, either with specified mentions to be disambiguated or without. Under the circumstance where mentions are not pointed out, Named Entity Recognition (NER) should be harnessed to finish the mention detection task. State-of-the-art NER methods utilize Neutral Networks and Deep Learning techniques to achieve better performances, whereas they have not been widely used yet on account of the freshness and complexity. Instead, Stanford NER Tagger, a NER tool which is less accurate but maturer, embraces higher popularity in tasks involving but not focusing on NER. In our experiment, we have already extracted the mentions during dataset construction process. This tagger is largely seen as the standard in named entity recognition, but since it uses an advanced statistical learning algorithm it's more computationally expensive than the option provided by NLTK.

Stanford NER is a Java implementation of a Named Entity Recognizer. Named Entity Recognition (NER) labels sequences of words in a text which are the names of things, such as person and company names, or gene and protein names. It comes with well-engineered feature extractors for Named Entity Recognition, and many options for defining feature extractors. It has a good named entity recognizers for English, particularly for the 3 classes (person,organization,location), and it also provide various other models for different languages and circumstances, including models trained on just the English training data.

A big benefit of the Stanford NER tagger is that is provides us with a few different models for pulling out named entities. We can use any of the following:

3 class model for recognizing locations, persons, and organizations
4 class model for recognizing locations, persons, organizations, and miscellaneous entities
7 class model for recognizing locations, persons, organizations, times, money, percents,dates.

Rules	Examples
Containment	Chicago $\rightarrow Chicagobulls$
Partial Matching	President Trump $\rightarrow DonaldTrump$ LA $\rightarrow LosAngeles$
Alternative Names	National Capital $\rightarrow WashingtonDC$

Table 3.1: String matching rules

After obtaining mentions, the following step is to retrieve possible candidate entities for each mention. Take Fig. 1.1 for instance, for mention Chicago, both entities Chicago Bulls and Chicago should be generated as candidates. In order to improve recall and generate more candidate entities, most KB-oriented EL methods tend to take advantage of name dictionaries embedded in KBs, or use alias dictionaries built from collecting Wikipedia redirecting and disambiguation pages. However, considering the limited number of target entities and sparse information of entity lists, we design a set of simple but efficient string matching rules for entity generation, as is shown in Table 3.1. In the examples, the left are mentions while the right are candidate entities.

The generated candidate entities for mention m_{ij} are represented by *Can*(M_{ij}). Note worthily, we adopt candidate pruning policy to ensure that a mention will not have two or more candidates from the same list, since entity list is utilized to help candidate entity within it to compete with entities from other lists, and choosing among candidate entities from the same list will render coherence within entity list useless.

3.1.2 Entity information enrichment

Solely relying on co-occurrences between entities is not enough to establish relations among entities, let alone semantically bridge mentions with candidate entities. Therefore, we enrich information on entity side by selecting representatives derived from input documents.

Given input documents D = $\{d_1, \ldots d_n\}$, the mentions $M_i = \{m_{i1}, \ldots m_{is}\}$ in each d_i, a set of entity lists $\varepsilon = \{E_1, \ldots E_l\}$, the enrichment process should collect a set of highly relevant and representative texts $T^r = \{t_1^r, \ldots t_h^r\}$ around mentions for E_r , which can be achieved by harnessing co-occurrences of entities in the same entity list.

Specically, the idea is that, since a document is not only composed of mentions, but also a lot of other irrelevant information, we merely extract the texts around all mentions in all documents as candidate representatives τ to avoid noisy information. If a candidate representative $t_p \in \tau$ contains many entity names from the same entity list E_r, chances are

that it indeed shares the same category or topic with entity list E_r, and the mention m_p in candidate representative t_p is thus much more likely to refer to the candidate entity from E_r. Consequently,t_p is a representative of E_r and the text in t_p can be used to enrich the textual descriptions of entities in E_r.

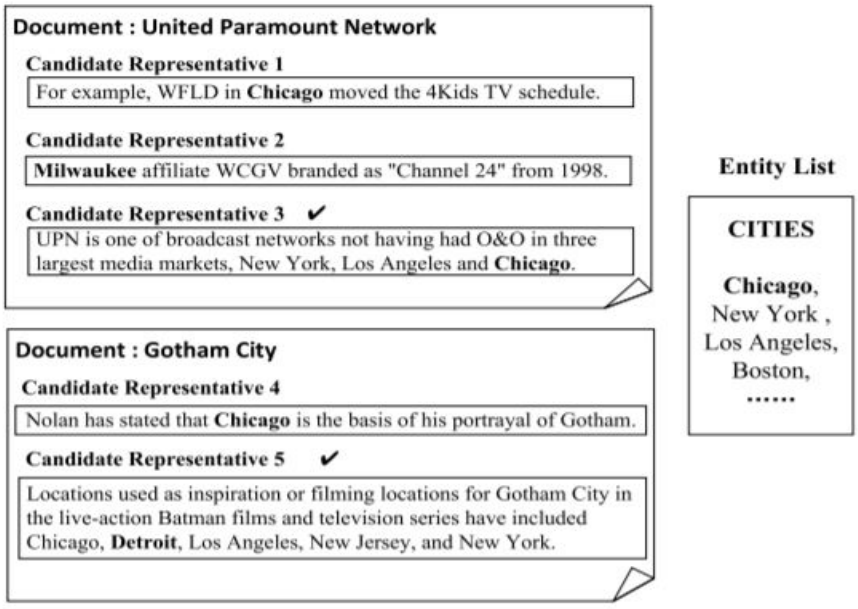


Figure 3.1: Example of entity information enrichment.

We further illustrate the method in Fig. 3.1. Note that in each candidate representative, the bold text represents a mention, and the rest texts are its surroundings. Given an entity list Cities and the entity Chicago, the goal is to collect relevant representatives for Chicago from documents, which are then used to enrich representatives of entity list Cities. In Document: United Paramount Network, there are three candidate representatives, two of them contain name string Chicago. However, Candidate Representative 1 includes no extra name strings of other entities from the entity list, thus might not refer to Entity List Cities. In contrary, both New York and Los Angeles co-occur with Chicago in Candidate Representative 3, indicating the high possibility that it is a true representative for Entity List Cities. Switching to Document: Gotham City, both Candidate Representatives contain name string Chicago. Despite the fact that Candidate Representative 4 is derived from mention Chicago, we cannot consider it as a representative due to lack of co-occurrences information. Conversely, containing several name strings from entity list Cities, Candidate Representative 5 is chosen as a representative, even though it is built surrounding mention Detroit.

3.2 Graph construction and disambiguation

In this subsection, we illustrate the construction of candidate entity graph, followed by the description of our proposed algorithm **Gloel**, which takes advantage of Personalized **Page Rank** so as to determine target entities.

3.2.1 Graph construction

Through the pre-processing step, mentions and their candidate entities are obtained. Then after enriching textual descriptions in the entity side, the compatibility score between each mention and corresponding candidate entity can be calculated in terms of text similarity. Previous listonly EL ranked the candidate entities for each mention merely based on mention-entity compatibility scores, thereby producing the results accordingly. We argue that the judgement simply depending on compatibility score is not convincing enough because the coherence among entities is ignored, which plays an indispensable role in the linking process. For instance, as is shown in Fig. 1.1, it is easy to map mentions Chicago and Los Angeles to the Cities entities Chicago and Los Angeles due to the short text information and high name string similarity. Provided that the candidate entity coherence is considered, the high interdependence among Teams entities New York Knicks, Chicago Bulls, Los Angeles Lakers would lead to the correct answers for mentions Chicago and Los Angeles.

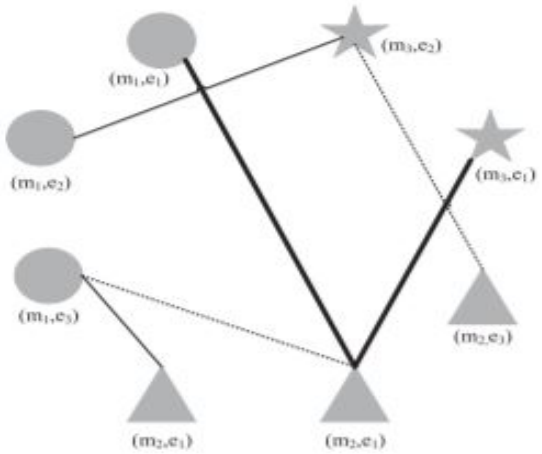


Figure 3.2: Entity graph.

To better capture the correlations among entities, similar to many existing KB-based EL methods, we construct an entity graph, which is depicted in Fig. 3.2. The definition of entity

graph is defined as follows

Definition 2 [Entity Graph] : An entity graph G = { V, E } is a weighed graph, in which the nodes V represent all candidate entities, with their source mentions specified, and edges E include relations between entities.

It is noteworthy that we differentiate the mentions with identical name strings even though they might appear in the same document, and similarly, by specifying the source mention of nodes, the candidate entities with the same name but generated from different mentions are also treated differently. In this way, the situations where there are duplicate nodes, either caused by mentions or entities, can be avoided. In the mathematical form, we represent the r-th candidate entity for mention m_{ij} in document d_i as $e_{ij,r}$, which clearly shows the source mention m_{ij} of candidate entity $e_{ij,r}$.

With reference to edges, following the tradition in KB-based EL and adapting it to list-only problem, we connect two nodes with an edge under three circumstances:

(1) The name strings of the two entities are in the same entity list $E \in \varepsilon$, and in this case, the edge weight is defined as 1.

(2) The name strings of the two entities simultaneously appear in at least one candidate representative $t \in \tau$.

(3) The name strings of the other entities in the entity lists these two entities separately belong to, simultaneously appear in at least one candidate representative $t \in \tau$.

The first two kinds of relations are termed as explicit relations, while the third method of adding edges among entities, named implicit relations mining, leverages the unique characteristic of entity list — that the rest entities $E' = E_i \{e_j\}$ in the same entity list E_i can help mine more correlations for entity e_j even if e_j is in the long tail. As for edge weight, which is defined below, takes into account both explicit and implicit relations between entities. Furthermore, the edges among candidate entities with the same source mention are pruned so as to eliminate the influence generated by competitors themselves. The existing method that we have so far,ignored the relation between entities and significance was given only to the mention - entity pair. By scoring the relation, the method explores the relation between entites also which was otherwise completely ignored.

We further assign initial node weight ini(v) and edge weight on the graph. The initial node weight ini(v) is defined as the compatibility score between candidate entity and its source

mention, while edge weight is determined by relation score between the entities on the two sides of the edge. The specific approaches to calculate compatibility score and relation score are given.

Compatibility score

Given a document d_i, and m_{ij}, a mention contained in d_i , suppose $e_{ij,r} \in E_r$ is a candidate entity for m_{ij} and $T^r = \{t_1^r, \ldots t_h^r\}$ is the set of representative texts for E_r . The compatibility score (m_{ij}, $e_{ij,r}$) can be measured by the following equation

$$ini(m_{ij},e_{ij,r}) = \phi(m_{ij},e_{ij,r}) = \frac{1}{T^r}\sum_{p=1}^{|T^r|} Sim\ (m_{ij},\ e_{ij,r}\ ) \qquad \text{(Equ:3.1)}$$

Since entities in the same entity list share the same representative texts, which are collected according to the method proposed in former section, calculating compatibility between a pair of mention m_{ij} and candidate entity $e_{ij,r} \in E_r$ can be converted to computing the average text similarity *Sim* between texts surrounding mention m_{ij} and all the text representatives T^r of candidate entity $e_{ij,r}$.

There are many ways to measure text similarity *Sim* and in this paper, we choose to compute the similarity between embedding vectors of two texts, which is represented as $E(m_{ij},t_p^r)$. Additionally, we also regard the name string similarity between mention and candidate entity as an appropriate indicator,and it is denoted as $N(m_{ij},e_{ij,r}$).Thus,the Compatibility score equation is converted to

$$\phi(m_{ij},e_{ij,r}) = (m_{ij},e_{ij,r}) + \beta\frac{1}{T^r}\sum_{p=1}^{|T^r|} E(m_{ij},\ e_{ij,r}\ ) \qquad \text{(Equ:3.2)}$$

In the equation above, α and β are the weight coefcients balancing the importance of text similarity and name string similarity.

In short, compatibility score gives us a measure of how close or related an entity is to its mention.It indicates why a particular entity is the right entity for the gives mention. It basically evaluates between mention and entity and does not provide an insight to the relation between two different entities in the context.

Relation score

The edge weight in the graph indicates the realtion score between any two entities. Given two entities $e_i^p \in E_p$, $e_j^q \in E_q$ (We merely consider relationships among entities when calculating Relation Score, which is mention-irrelevant, thus we neglect the mention here), the Relation Score is denoted in the following equation

$$\text{Rel}(e_i^p, e_j^q) = \begin{cases} \eta O(e_i^p, e_j^q) + \frac{\theta}{M} \sum_u^{E-p-i} \sum_v^{E-q-j} O(e_u^p, e_v^q) \\ 1 \end{cases} \quad \text{(Equ:3.3)}$$

where $\text{O}(e_i^p, e_j^q) = \frac{|occur(e_i) \cap occur(e_j)|}{|occur(e_i) \cup occur(e_j)|}$

M = $(|E_p| - 1)(|E_q| - 1)$

We illustrate equations above as follows : occur(e) denotes the occurrences of entity e in all candidate representatives , since compared with noisy textual information contained in the whole documents, merely considering texts around mentions (candidate representatives) can improve the accuracy.

The Co-occurrence Frequency O(e_i, e_j) of two entities e_i and e_j is defined as the number of candidate representatives they both occur in, divided by all the candidate representatives they either occur in together, or separately. As for the Relation Score Rel(e_i^p, e_j^q) of two entities $e_i^p \in E_p$, $e_j^q \in E_q$, if p equals q, which means e_i^p and e_j^q are from the same entity list, we set the relation score as 1. Otherwise, the score is composed of two parts. The first component is the direct Co-occurrence Frequency of these two entities, multiplied by a weight factor , which indicates explicit relations. The implicit relations are represented by indirect Co-occurrence Frequency, which is the second component with a coefficient , and it takes into account the co-occurrences of the rest entities in E_p and E_q in a pair-wise fashion.

Furthermore, as is shown in Fig. 3.2, there are three kinds of lines. The bold line represents that entities on the two sides are in the same entity list, and the Relation Score is 1. The dotted line denotes that two entities merely have implicit relations, while the normal line requires that there are explicit relations between entities.

It is noteworthy that, different from traditional KB-oriented EL problem which merely considers the direct relations between two entities, we extend the definition by taking into account the contribution made by relations between two entity lists as well, and represent them as implicit relations of two entities. The detailed approach to quantitatively describe

the implicit relations is embodied in the equations above.

3.2.2 Ranking mention-entity pairs

Given a weighed entity graph G_i of document d_i, the target is to find the most likely entity, $e_{ij,k}$ from a group of entities for each mention m_{ij} in document d_i. In line with popular methods proposed in KB-oriented EL [3], we propose graph-based list-only entity linking algorithm, namely **Gloel**, which utilizes Personalized **PageRank** to depict the coherence among candidate entities.

Specifically, we assign a vector $p(v_s)$ with length n to each node v_s to represent the results of a **PageRank** process starting from v_s. To better capture the coherence among entities within the same document, instead of regarding the similarity between the vectors of nodes as the coherence score, we define it as how a candidate entity fits in the document. To enable the definition, a n-length vector $p(d_i)$ is also assigned to document d_i, representing the results of the **PageRank** process initiating from a group of unambiguous nodes. Consequently, the coherence score of a candidate entity $e_{ij,r}$ for mention m_{ij} in document d_i is defined as

$$\psi(e_{ij,r},d_i) = \frac{p(v_{ij,r})p(d_i)}{|p(v_{ij,r})||p(d_i)|} \quad \text{(Equ:3.4)}$$

We first elaborate the random walk process initiating from a single node, then extend it to calculating document **PageRank** vector. The **PageRank** algorithm, based on random walk theory, is firstly proposed to measure the importance of web pages by counting the number and quality of links to this page. It has been applied to EL problems in recent years, and has achieved great performance.

The notion of PageRank, first introduced by Sergery Brin and Larry Page, forms the basis for their Web search algorithms. Although the original version of PageRank was used for the Webgraph (with all the webpages as vertices and hyperlinks as edges), PageRank is well defined for any given graph and is quite effective for capturing various relations among vertices of graphs. it has been found that there are several implications of PageRank for a given graph. To start with, we give the graph-theoretical definition of PageRank. To begin with, PageRank is a way to organize random walk of various lengths. Instead of having to determine the number of steps a random walks is taking, PageRank uses a positive real value α, where $\alpha \in [0, 1)$ to control the "diffusion" of a combination of random walks.

The original definition for PageRank was to assign a value to each vertex (Webpage), denoting the importance of a vertex under two assumptions: For some fixed probability α,

a surfer at a Webpage jumps to a random Webpage with probability and goes to a linked Webpage with probability 1 - α. The importance of a Webpage v is the expected sum of the importance of all the Webpages u that preceed v.

The basic elements of **PageRank** include initial vector r_0, transition matrix A, and preference vector s. Note that in our method, r_0= s. Transition Matrix A is the same in both individual and collective processes, the value at i^{th} row and j^{th} column is defined as

$$A_{ij} = \frac{Rel(e_i, e_j)}{\sum_{e_k \in Edges(e_i)} Rel(e_i, e_k)} \quad \text{(Equ:3.5)}$$

where Edges(e_i) represents the edges connected to entity e_i . When computing the vector p(v_t) for a single node v_t, r^0 = s = (0 . . . 0, 1(*t*th), 0 . . .)$_n$, which means that r^0 and s are identical n-length vectors, the position t of the vector is assigned with 1 and the rest are endowed with 0.

The situation is slightly more complicated as for document PageRank vector p(d_i). Firstly, we regard a candidate entity $e_{ij,r}$,r as a unambiguous one if it satisfies one of the following conditions:

1) $e_{ij,r}$ is the only candidate entity of mention m_{ij} and ini($e_{ij,r}$) is above threshold μ. The unambiguous entities of this kind is endowed with initial weight λ.

2) When there are more than one candidate entities and $e_{ij,r}$ is the candidate entity with the largest initial value, suppose ($e'_{ij,r}$) is the candidate entity with the second largest initial value. It suffices that ini($e_{ij,r}$) ini($e'_{ij,r}$) $\geq \nu$. The initial weight of this kind is κ.

3) If there are no candidate entities meeting the conditions, all the candidate entities will be added to the unambiguous entities set, with the same weight endowments.

After obtaining unambiguous entities set, the actual weight can be assigned via normalization of initial weight values. Note that in graph, unambiguous entities are presented as equivalent nodes, and by placing the actual weight of unambiguous nodes in the corresponding positions of the n-length vector, we can attain r^0 and s for document accordingly. Furthermore, we adopt an iterative disambiguation approach. In other words, after erasing ambiguity for each mention, the chosen result entity will be regarded as unambiguous and added to the unambiguous entities set, with initial weight of . Afterwards, the document **PageRank** vector will be re-computed by utilizing the new unambiguous entities set. With initial vector r^0, transition matrix A, and preference vector s defined as above, the Personalized **PageRank** is

presented as following

$$\gamma^{t+1} = (1 - \rho) \times A \times r^t + \rho \times s \qquad \text{(Equ:3.6)}$$

In the equation 3.6, t represents the t^{th} iteration, and denotes the probability that the random walk process jumps out of the original iteration and starts from a new vector, which is usually set at 0.15. Normally, the restarting nodes are all nodes in the graph, and the weights in vector s are the same, which equal to 1/|V|. Nonetheless, in this work, vector s is personalized and set as the same with initial vector, which means that the random walk merely restarts from the initial nodes, eliminating the effect from other nodes. When the iterative calculation reaches to a stage where r^k does not change any more or the variation is within a minimal range, we consider that it converges and $p(v_s)$, $p(d_i)$ are thereby attained. At last, we formalize the list-only EL problem in a mathematical way :

Definition 3 [List-Only Entity Linking in Mathematical Form] : Given a set of documents D = $\{d_1, \ldots d_n\}$, each of which contains a set of mentions $M_i = \{m_{i1}, \ldots m_{is}\}$, an ambiguous set of entity lists $\varepsilon = \{E_1, \ldots E_l\}$, the task is to determine the most possible entity $e_{ij,k} \in E_k$ for each mention m_{ij}, and the chosen entity is given by the following equation

$$e_{ij,k} = \arg\max_{e_{ij,r} \in Can(m_{ij})} (\gamma\phi(m_{ij}, e_{ij,r}) + \delta\psi(e_{ij,r}, d_i)) \qquad \text{(Equ:3.7)}$$

where γ and δ are two weight coefficients balancing the weight between mention-entity compatibility score (from Equ:3.2) and entity coherence score (from Equ:3.4). NIL will be returned if there is no corresponding entity.

3.3 Dataset selection

The current list-only EL dataset contains 11065 documents and 7 groups of entity lists, with 139 entities in total. Each document merely includes a single mention to be disambiguated. In addition, the entity lists cover the categories of President, Company, University, State, Character, Brand, Restaurant, and the entities in different entity lists are disparate both in terms of surface forms and true meanings.

Category	Name Examples
President	Barack Obama, Ronald Reagan
Company	Microsoft, Apple, Adobe, IBM
University	Harvard University, Yale University
State	Washington, Florida, California, Texas
Character	Gandalf, Aragorn, Legolas, Gimli, Frodo
Brand	Prada, Chanel, Burberry, Gucci, Cartier
Restaurant	Subway, McDonald's, KFC, Starbucks

Figure 3.3: Data set

There are two shortcomings in current dataset. For one thing, each document merely contains a single mention to be disambiguated, which does not fit in most real-life occasions. For another, the target entity lists are not ambiguous enough, giving rise to the situation that most mentions merely have one candidate entity, and the disambiguation problem is converted to judging whether this sole candidate entity is true or not. Take entity Apple in Company entity list shown in the dataset , there is no other similar entities in the set of entity lists. As a result, when given a mention Apple, the candidate entity for it will only be Apple in Company entity list, and the problem is transformed into deciding whether the mention can be mapped to entity lists or not.

In order to overcome the deficiencies, we propose to mine target entity lists and collect documents. The entity lists can be constructed both manually and automatically, but the ambiguity must be ensured. Given two entity lists E_m = { $e_{1,m}$, . . . $e_{i,m}$ } and E_n = { $e_{1,n}$, . . . $e_{j,n}$ } for E_m, the ambiguity caused by the existence of E_n is defined as

$$\text{Amb}(E_m, E_n) = \frac{1}{|E_m|}\sum_{e_{i,m}\in E_m} \arg\max_{e_{j,n} \in Can(m_{ij})} \text{amb}(e_{im}, e_{jn}) \qquad \text{(Equ:3.8)}$$

Note that *amb(e_{im}, e_{jn})* represents the ambiguity between two entities in different entity lists. Many approaches can be utilized to measure it, and in this paper, we harness the matching rules defined in the candidate entities retrieval section. If matching rules are satisfied, we endow 1 to *amb(e_{im}, e_{jn})*. Otherwise, the value is determined by name string similarity. Furthermore, the reason why only the highest ambiguity value for e_{im} is chosen lies in the fact that we merely need to assure e_{im} has one ambiguous competitor to avoid the situation as the example above As for building the documents dataset, we emphasize that there have to be at least two mentions in the same document to enable the construction of candidate entity graph. Otherwise there will be no difference between independent linking method and the proposed collective linking method based on graph. To be specific, we utilized wikilinks

in Wikipedia to obtain the documents. For each entity ei,k in entity list Ek , its referent Wikipedia page was determined in the first place. For instance, the Wikipedia page of entity Atlanta is *en.wikipedia.org/wiki/Atlanta*. Then we randomly retrieved 1,000 Wikipedia pages directing at ei,k via the *WhatLinksHere* page. As for Atlanta, the url of its *WhatLinksHere* page is *en.wikipedia.org/wiki/Special: WhatLinksHere/Atlanta*. After conducting the same operation for all the entities in entity list E_k , the links appearing in at least three entities' 1,000 Wikipedia pages were selected and the web pages texts they refer to were considered as documents. In this way, we can affirm that each document involves at least three mentions.

3.4 Performance evaluation

We compare Gloel and the method utilized so far (denoted as Independent) on the dataset we create. The settings of parameters are listed as follows: $\alpha = 0.4$, $\beta = 0.6$, $\eta = 0.7$, $\theta = 0.3$, $\gamma = 0.5$, $\delta = 0.5$, $\lambda = 0.5$, $\kappa = 0.4$.

The measurements we adopt are the same, namely Precision, Recall and F1. Precision takes into account all entity mentions that are linked by the system and determines the correctness. Recall on the other hand, considers all the mentions should be linked, and reflects the fraction of correctly linked mentions. F1 is a balanced indicator of Precision and Recall.

Gloel outperforms independent EL method in all occasions, with a overall F1 gain at 1.1%. Nevertheless, it is evident that both methods achieve high Precision, Recall and F1 scores. This can be justified that most mentions in the documents appear in the same name string form as the entity name strings. For instance, in documents containing mention referring to entity University of Cambridge, the name form of the mention is also University of Cambridge, thus the high name string similarity basically guarantees the correct matching and rules out the possibility of other candidate entities. Plus, this does not fit in situations of most text sources other than Wikipedia. In news reports concerning University of Cambridge, it constantly goes by the name Cambridge as in sentence Cambridge beats Oxford in terms of computer science. In these cases, the probability of generating result entity Cambridge is enhanced significantly and the disambiguation difficulty also rises up.

As a consequence, we corrupted the dataset to observe the corresponding results produced by these two methods. For instance,the mention names Atlanta Hawks and Atlanta Braves were substituted by Atlanta. Considering the fact that after corruption, the name string similarity might be of no use and possibly lead to negative contributions, which was unfair for Independent results, we merely took into account the embedding vectors similarity in terms of mention-entity similarity calculation and altered the corresponding parameter setting. It is noteworthy that, for each corruption degree, we generated five corrupted corpus and reported

the average results so as to increase the stability and persuasiveness of the outcomes.

The gap between the results of Independent and Gloel widens after the data had been 50% corrupted. Gloel achieves better outcomes with overall F1 score at 90.9%, while the overall F1 value of previous method is 76.5%, hence validating the superiority of the proposed method. We further conducted 25% and 75% corruption on the dataset and Fig. 3.2 dynamically depicts the F1 scores of these two methods under corruption. With input texts getting more difficult, the results of EL based solely on mention-entity compatibility decline rapidly, while Gloel, a method based on graph, still yields robust results with smaller decreases.

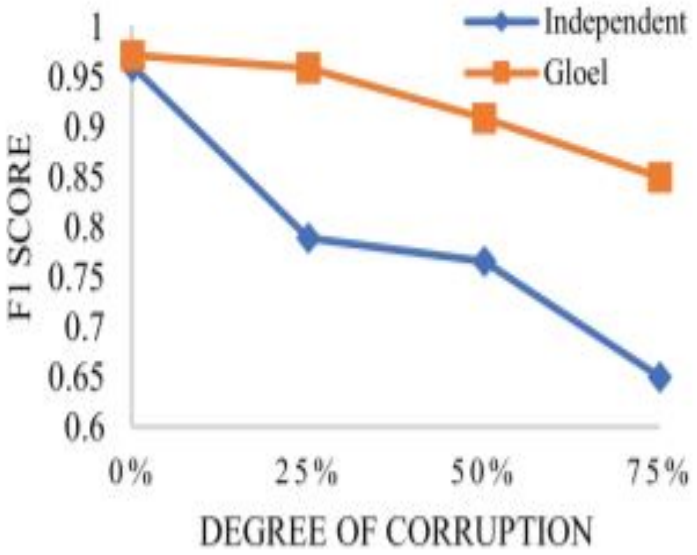


Figure 3.4: F1 score of Independent and Gloel over corrupted dataset.

Conclusion

The List-only entity linking task, as a new form of traditional EL problem, distinguishes itself by the sparse information on the entity side. In this work, on the one hand, we propose to utilize entity co-occurrences information to mine both textual description of entities and relations among entities, so as to enrich entity information. On the other hand, inspired by conventional EL methods, we construct an entity graph to capture relations among entities, on which the newly proposed algorithm Gloel is applied to obtain results. Similar to the situation in traditional EL, this approach, a collective EL method based on graph, outperforms independent EL on the dataset created for fair comparison.

For future work, it is planned to investigate two aspects. One is to consider the situation where an entity appears in more than one entity list. For instance, Washington, D.C. can appear in entity lists featured American Cities and Country Capitals.

Another possible research direction is utilizing word embedding techniques and deep neural networks to better model mention-entity compatibility and entity coherence. Specifically, leveraging well-trained word embedding vectors as inputs, Long Short-Term Memory (LSTM) [7] with attention mechanism could be used to summarize semantic meanings of the contexts around mentions and the representative texts of entities, which can be further harnessed to calculate more accurate compatibility score.

References

[1] Y. Lin, C.-Y. Lin, and H. Ji, "List-only entity linking," in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, Vancouver, BC, Canada, 2017, pp. 536–541. [Online]. Available: https://doi.org/10.18653/v1/P17-2085

[2] Y. Cao, J. Li, X. Guo, S. Bai, H. Ji, and J. Tang, "Name list only? Target entity disambiguation in short texts," in Proc. Conf. Empirical Methods Natural Lang. Process., Lisbon, Portugal, 2015, pp. 654–664. [Online]. Available: http://aclweb.org/anthology/D/D15/D15-1077.pdf

[3] Z. Guo and D. Barbosa, "Robust entity linking via random walks," in Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage., Shanghai, China, 2014, pp. 499–508. [Online]. Available: http://doi.acm.org/10.1145/ 2661829.2661887

[4] X. Han, L. Sun, and J. Zhao, "Collective entity linking in Web text: A graph-based method," in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., Beijing, China, 2011, pp. 765–774. [Online]. Available:http://doi.acm.org/10.1145/2009916.2010019

[5] A. Alhelbawy and R. J. Gaizauskas, "Graph ranking for collective named entity disambiguation," in Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Baltimore, MD, USA, 2014, pp. 75–80. [Online]. Available: http://aclweb.org/anthology/P/P14/P14-2013.pdf

[6] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., Denver, CO, USA, 2015, pp. 238–243. [Online]. Available: http://aclweb.org/ anthology/N/N15/N15-1026.pdf

[7] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., Denver, CO, USA, 2015, pp. 238–243. [Online]. Available: http://aclweb.org/ anthology/N/N15/N15-1026.pdf

[8] C. Wang, K. Chakrabarti, T. Cheng, and S. Chaudhuri, "Targeted disambiguation of ad-hoc, homogeneous sets of named entities," in Proc. 21st World Wide Web Conf., Lyon, France, 2012, pp. 719–728. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187934