List-only Entity Linking

Ying Lin[1] [*], **Chin-Yew Lin**[2], **Heng Ji**[1]
[1] Computer Science Department,
Rensselaer Polytechnic Institute, Troy, NY, USA
{liny9,jih}@rpi.edu
[2] Microsoft Research, Beijing, China
cyl@microsoft.com

Abstract

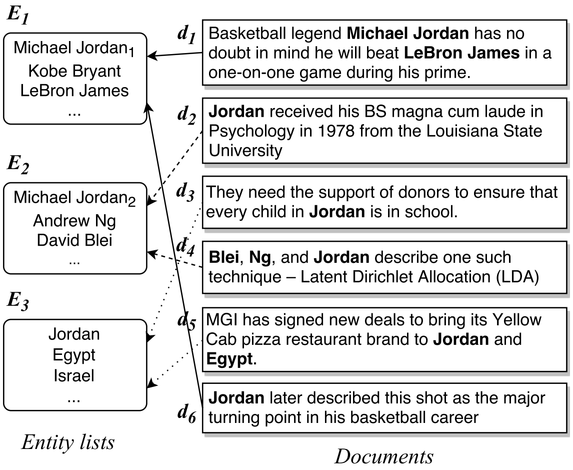
Traditional Entity Linking (EL) technologies rely on rich structures and properties in the target knowledge base (KB). However, in many applications, the KB may be as simple and sparse as lists of names of the same type (e.g., lists of products). We call it as *List-only Entity Linking* problem. Fortunately, 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. Our experiments on linking mentions to seven automatically mined lists show promising results and demonstrate the effectiveness of our approach.[1]

1 Introduction

Traditional Entity Linking (EL) methods usually rely on rich structures and properties in the target knowledge base (KB). These methods may not be effective in applications where detailed descriptions and properties of target entities are absent in the KB. Consider the following situations:

Disaster Response and Recovery. When a disaster strikes, people rush to the web and post tweets about the damage and casualties. Performing EL to extract key information, such as devastated towns and donor agencies, can help us monitor the situation and coordinate rescue and recovery efforts. Although many involved entities are not well-known and usually absent in general KBs, we may be able to acquire lists of these entities from the local government as the target KB.

Voice of the Customer. EL also plays an important role in mining customer opinions from data generated on social platforms and e-commerce websites, thereby helping companies better understand the needs and expectations of their customers. However, the target products are often not covered by general KBs. For example, (Cao et al., 2015) tested 32 names of General Motors car models and only found 4 in Wikipedia. Although some companies may choose to maintain a comprehensive product KB, it will be much more practical and less costly to provide only lists of product names.



Basketball legend **Michael Jordan** has no doubt in mind he will beat **LeBron James** in a one-on-one game during his prime.

Jordan received his BS magna cum laude in Psychology in 1978 from the Louisiana State University

They need the support of donors to ensure that every child in **Jordan** is in school.

Blei, **Ng**, and **Jordan** describe one such technique – Latent Dirichlet Allocation (LDA)

MGI has signed new deals to bring its Yellow Cab pizza restaurant brand to **Jordan** and **Egypt**.

Jordan later described this shot as the major turning point in his basketball career

Figure 1: Link mentions to target entities in different entity lists.

Under such circumstances, we need the ability to perform EL to ad-hoc name lists instead of a comprehensive KB, namely List-only Entity Linking. Take Figure 1 as an example. For a human reader, it is not difficult to figure out the referent entities of mentions in each document based on

[*] Part of this work was done when the first author was on an internship at Microsoft Research Asia.

[1] The data set is available at: http://nlp.cs.rpi.edu/data/link-only-entity-linking.html

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 536–541
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-2085

clues such as "basketball" and "LDA", whereas we will not be able to make such inference without the knowledge of the target entities. However, even if we lack the minimal knowledge (*e.g.*, `Jordan` is a country), we are more confident to link mentions in d_1, d_4, and d_5 because they co-occur with other entities in the same list. We consider such mentions that we are confident to link as *seed mentions*, and use them to construct contextual and non-contextual information of the target entities to enhance entity disambiguation.

Therefore, in this work, we propose to tackle the problem of List-only Entity Linking through *seed mentions*. We automatically identify seed mentions for each list using a two-step method based on the occurrence of entities and similarity between mentions. After that, in the entity disambiguation phase, we utilize the selected mentions as a bridge between uninformative entities and other mentions. Specifically, we comparing features of a non-seed mention to those of seed mentions of its entity candidates to determine which entity it should be linked to.

2 Problem Definition

Given a mention m and the entity e that it refers to, we call e the *referent entity* of m and m the *referential mention* of e. In Figure 1, for example, `Michael Jordan`$_1$ is the referent entity of "Jordan" in document d_6, while "Jordan" in document d_2 is an non-referential mention for `Michael Jordan`$_1$.

As Figure 1 shows, in the setting of List-only Entity Linking, there are a set of manually or automatically generated entity lists $\mathcal{E} = \{E_1, E_2, ..., E_l\}$ and a set of documents $D = \{d_1, d_2, ..., d_n\}$. Entities in the same list are homogeneous and share some common properties. In our experiment, each document d_i contains a mention m_i to link. Our goal is to link m_i to its referent entity $e_{i,j} \in E_j$ or returns NIL if it is unlinkable to any entities.

3 Approach

Our framework has two modules, *entity candidate retrieval* and *entity disambiguation* as Figure 2 shows. For a mention "Jordan," we retrieve two candidate entities, `Michael Jordan`$_1$ and `Michael Jordan`$_2$, from the entity lists. Next, we select a set of seed mentions for each entity from all documents. To determine the referent entity of "Jordan", we compare it with seed mentions of each candidate instead of the entity itself.

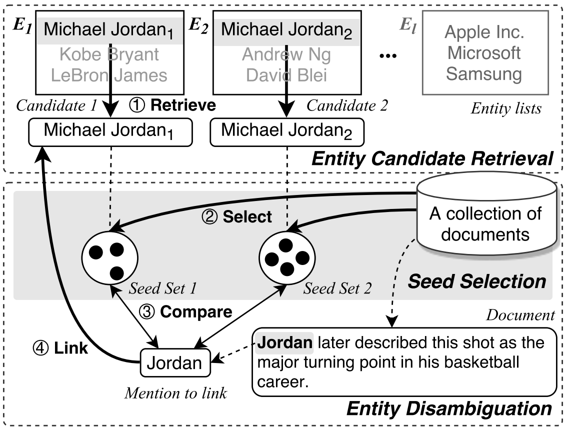


Figure 2: List-only Entity Linking Framework.

3.1 Entity Candidate Retrieval

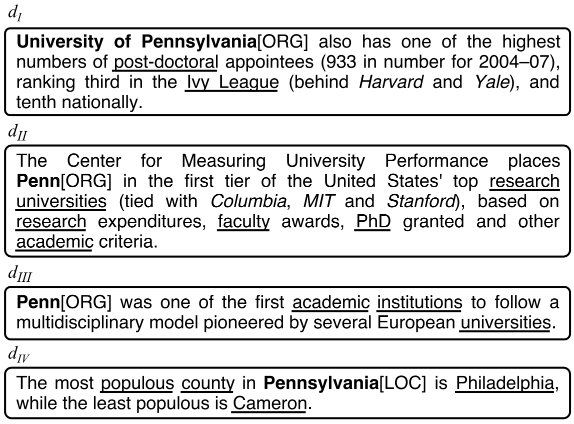
For each mention m_i, we first locate a set of entity candidates $C_i = \{e_{i,j} | e_{i,j} \in E_j\}$ that it possibly refers to. A mention and its referent entity may have different surface forms (*e.g.*, "BMW" and `Bayerische Motoren Werke`). For this reason, we design a set of matching rules to improve the recall as shown in Table 1.

Category	Rule	Examples
Abbreviation	Acronym	USDOD/USDD (United States Department of Defense),
	Initial Letters	corp. (corporation), univ. (university)
	First and Last Letters	Dr. (Doctor), PA (Pennsylvania)
	Omission	Address a person by his/her given name or surname rather than full name
Substitution	Numeral	7-11 (7-Eleven)
	Symbol	AT&T (American Telephone and Telegraph Company)
	Accent Mark	hermes.com (Hermès)

Table 1: Alternative form matching rules.

3.2 Entity Disambiguation

Next, we proceed to score each candidate $e_{i,j}$ and determine which one m_i should be linked to. However, we have no knowledge of the target entities except for names and thus can't directly compare m_i with them. Rather, we propose to bridge the gap between mentions and entities through *seed mentions*.



d_I

University of Pennsylvania[ORG] also has one of the highest numbers of post-doctoral appointees (933 in number for 2004–07), ranking third in the Ivy League (behind *Harvard* and *Yale*), and tenth nationally.

d_{II}

The Center for Measuring University Performance places **Penn**[ORG] in the first tier of the United States' top research universities (tied with *Columbia*, *MIT* and *Stanford*), based on research expenditures, faculty awards, PhD granted and other academic criteria.

d_{III}

Penn[ORG] was one of the first academic institutions to follow a multidisciplinary model pioneered by several European universities.

d_{IV}

The most populous county in **Pennsylvania**[LOC] is Philadelphia, while the least populous is Cameron.

Figure 3: Bridging the gap between uncertain mentions and target entities using seed mentions.

We illustrate the idea in Figure 3. `University of Pennsylvania` is retrieved as an entity candidate for mentions "University of Pennsylvania", "Penn", and "Pennsylvania." We are more confident to link "University of Pennsylvania" in d_I and "Penn" in d_{II} to `University of Pennsylvania` because other entities in the *University* list, such as "Harvard" and "MIT," also appear in the same document. Thus, we select mentions in d_I and d_{II} as seed mentions. From d_I and d_{II}, we can extract both contextual features (*e.g.*, "academic" and "research") and non-contextual features (*e.g.*, the entity type is ORG). After that, we compare mentions in other documents with the seeds. We link "Penn" in d_{III} to `University of Pennsylvania` because its entity type and context are consistent with the seeds. "Pennsylvania" in d_{IV}, however, is not linked because it is recognized as a location. To capture richer contextual information and minimize the effect of noise, we select more than one seed mention using a two-step approach as follows.

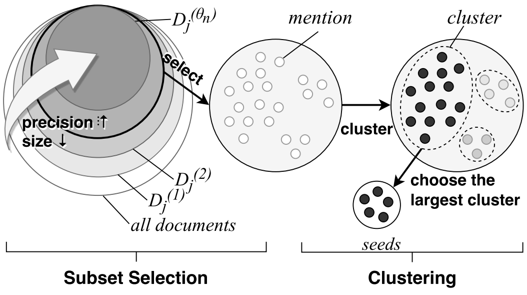


Figure 4: Seed selection.

1. **Subset Selection**. We assume that if multiple names in the same list co-occur within a document, they are all likely to be referential mentions of this list, such as "Michael Jordan" and "LeBron James" in d_1. Hence, to identify seed mentions of list E_j, we first narrow the scope down to a subset $D_j^{(n)}$ of documents containing more than n mentions matching names in E_j. We gradually increase the n until $\theta^-_{\text{size}} \leq |D_j^{(n)}| \leq \theta^+_{\text{size}}$. θ^-_{size} and θ^+_{size} are set to 50 and 300 in our experiments.

2. **Clustering**. We expect most mentions in the selected subset are referential of list E_j, while in fact the subset is likely to contain a small number of non-referential mentions. We need to eliminate them from the subset, otherwise they will introduce misleading features differing from the real seed mentions, hence hurting the performance of entity disambiguation. To separate referential and non-referential mentions in the selected subset, we make two assumptions: (1) Most mentions in the subset are referential, and (2) Referential mentions should be similar to each other while dissimilar from non-referential ones. Due to the lack of annotated data, we approach this problem by performing clustering, which works in an unsupervised fashion. Specifically, we represent features (described later in this section) of each mention as a vector and measure the distance between two mentions using cosine distance. After that, we run the K-means++ algorithm on the subset to separate referential and non-referential mentions, and pick mentions in the largest cluster as seed mentions.

To determine the referent entity of mention m_i, we calculate the confidence score of linking m_i to $e_{i,j} \in E_j$ using the average cosine similarity between m_i and seed mentions of list E_j:

$$c(m_i, e_{i,j}) = \frac{1}{|S_j|} \sum_{p=1}^{|S_j|} \text{sim}(m_i, m_s), m_s \in S_j$$

where S_j is the seed set of E_j. Lastly, we link m_i to the candidate with the highest confidence score.

In this work, we use the following features.

Entity Type. The entity type of a mention can be inferred from the text and used for disambiguation. For example, if most seed mentions for the *University* list are recognized as ORG, while "Harvard" in the sentence "Harvard was born and raised in Southwark, Surrey, England" is tagged as PER, it is unlikely to refer to `Harvard University`.

Textual Context. We also assume that referential mentions of the same entity should share similar local contexts. We represent textual context using the average embedding of words within a window around the mention.

Punctuation. Punctuations preceding or following a mention may help resolve ambiguity. For example, “MA” preceded by a comma is possible to refer to a state, since states are usually the last component of an address, such as “Boston, MA”.

4 Experiments

4.1 Data set

In our experiment, the construction of data set consists of two steps: collecting name lists from NeedleSeek[2] (Shi et al., 2010) and extracting documents from Wikipedia. NeedleSeek is a project aiming to mine semantic concepts from tera-scale data (ClueWeb09) and classify them into a wide range of semantic categories. For example, “KFC” is mined as a concept in the restaurant category, along with key sentences and attributes, such as employee number and founder.

To obtain target name lists, we select 7 semantic categories (see Table 2) generated by NeedleSeek as target domains, and take the top concepts in each category as target entities. We manually map each name to its pertinent Wikipedia page as a target entity (e.g., Starbucks → `enwiki:Starbucks`[3]). Thus, we collect lists containing 139 target entities in total. Note that category names are only for result presentation purpose and not taken as input to our model.

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

Table 2: Semantic categories from NeedleSeek.

Next, we derive a data set from Wikipedia articles through wikilinks[4], which are links to pages within English Wikipedia. For example, a wikilink `[[Harvard University|Harvard]]` appears as “Harvard” in text and links to the page `enwiki:Harvard_University`. Thus, we can consider “Harvard” as a name mention and `enwiki:Harvard_University` as its referent entity. Consider the following sentences:

∗ ... *then left toattend graduate school on a scholarship at* `[[Harvard University|Harvard University]]`...

∗ *On October 6, 2012,* `[[Allison Harvard|Harvard]]` *made an appearance in an episode of...*

Because `enwiki:Harvard_University` is in the *University* list, the first mention will be considered as referential, whereas the second one is non-referential. We also apply matching rules in Table 1 to obtain more non-referential mentions. After that, we extract sentences around wikilinks as a document.

Category	#Referential	#Referential (balanced)	#Non-referential
President	51, 412	14, 722	14, 818
Company	13, 312	3, 604	3, 642
University	79, 285	30, 101	30, 187
State	86, 743	9, 602	9, 106
Character	729	483	476
Brand	5, 138	1, 739	1, 781
Restaurant	4, 261	4, 261	4, 850
Total	240, 588	64, 512	61, 632

Table 3: Data set stats.

From Table 3, we can see that referential entities overwhelm non-referential ones in the extracted corpus. In order to evaluate our model fairly, we perform downsampling to balance referential and non-referential mentions, otherwise we can achieve high scores even if we link all mention to the target entities. In the balanced data set, there are $11,065$ unique entities.

4.2 Entity Linking Results

Category	Complete			Balanced Subset		
	R	P	F	R	P	F
President	94.6	89.9	92.2	87.2	80.4	83.7
Company	86.6	95.8	91.0	90.8	85.1	87.9
University	96.7	96.4	96.5	96.9	92.0	94.4
State	96.2	92.1	94.1	95.0	58.6	72.5
Character	92.5	61.3	73.7	92.8	52.2	66.8
Brand	89.6	90.2	89.9	86.7	83.2	84.9
Restaurant	87.0	81.4	84.1	86.9	88.1	87.5
Overall	95.2	93.4	94.3	93.1	81.6	87.0

Table 4: Overall performance (%). R, P, and F represent recall, precision, and F1 score, respectively.

As Table 4 demonstrates, our method shows promising results (87.0 F1 score) on the balanced data set. Nevertheless, we notice the low linking precisions for entities in the *Character* and *State* lists, which are caused by different reasons. For the *Character* list, mentions do not suffice to select

[2] http://needleseek.msra.cn

[3] `enwiki:` is short for https://en.wikipedia.org/wiki/

[4] https://en.wikipedia.org/wiki/Help:Link

high-quality seeds, whereas for the *State* list, features of referential and non-referential mentions are usually similar. Consider the following sentence:

∗ *She witnessed his fatal shooting when they were together in the President's Box at Ford's Theatre on Tenth Street in Washington.*

The mention "Washington" refers to "`Washington, D.C.`", which has the same entity type, LOCATION, as our target entity "`Washington (state)`". In addition, we see no obvious textual clue that indicates whether it refers to the State of Washington or not. Traditional EL approaches usually disambiguate such mentions through collective inference. They link "Ford's Theatre" and "Washington" to the KB simultaneously. Since there exists an explicit relation between "`Ford's Theatre`" and "`Washington, D.C.`", these two entities receive high confidence scores and thus are determined as the referents. Unfortunately, we cannot employ the knowledge-rich approach in the List-only Entity Linking scenario.

5 Related Work

In this paper, we define and study the List-only Entity Linking problem based on previous studies on Target Entity Disambiguation (Wang et al., 2012; Cao et al., 2015). The key difference is that they target at the disambiguation of a single list of entities, whereas we focus on entity linking to an arbitrary number of lists. Another similar problem is Named Entity Disambiguation with Linkless Knowledge Bases (LNED) (Li et al., 2016). It assumes that entities are isolated in the "linkless" KB, while each entity still has a description.

Our idea of selecting seed mentions based on co-occurrence is similar to collective inference. Most state-of-the-art EL methods utilize collective inference to link a set of coherent mentions simultaneously by selecting the most coherent set of entity candidates on the KB side (Pan et al., 2015; Huang et al., 2014; Cheng and Roth, 2013; Cassidy et al., 2012; Xu et al., 2012). In this work, without explicit relations between entities in different lists, we only take the co-occurrence of mentions in the same list into consideration. Therefore, our method is unable to benefit from the co-occurrence of `John Lennon` and `Give Peace a Chance` although they are actually strongly connected.

6 Conclusions and Future Work

In this paper, we proposed a novel framework to tackle the problem of List-only Entity Linking. The core of this framework is selecting seed mentions for each entity list to bridge the gap between mentions and non-informative target entities. Our results show this EL framework works well for this task. At present, in the seed selection step, we simply consider all co-occurring mentions of entities in the same list. In the future, we will employ more precise approaches to choose co-occurring mentions and mine relations between entities in separate lists to improve seed selection and entity disambiguation.

Acknowledgments

This work was supported by the DARPA DEFT No.FA8750-13-2-0041, U.S. DARPA LORELEI Program No. HR0011-15-C-0115, U.S. ARL NS-CTA No. W911NF-09-2-0053, and NSF IIS-1523198. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Yixin Cao, Juanzi Li, Xiaofei Guo, Shuanhu Bai, Heng Ji, and Jie Tang. 2015. Name list only? target entity disambiguation in short texts. In *EMNLP*. https://doi.org/10.18653/v1/D15-1077.

Taylor Cassidy, Heng Ji, Lev-Arie Ratinov, Arkaitz Zubiaga, and Hongzhao Huang. 2012. Analysis and enhancement of wikification for microblogs with context expansion. In *COLING*. http://aclweb.org/anthology/C12-1028.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *EMNLP*. http://aclweb.org/anthology/D13-1184.

Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. 2014. Collective tweet wikification based on semi-supervised graph regularization. In *ACL*. https://doi.org/10.3115/v1/P14-1036.

Yang Li, Shulong Tan, Huan Sun, Jiawei Han, Dan Roth, and Xifeng Yan. 2016. Entity disambiguation with linkless knowledge bases. In *WWW*.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In *NAACL*. https://doi.org/10.3115/v1/N15-1119.

Shuming Shi, Huibin Zhang, Xiaojie Yuan, and Ji-Rong Wen. 2010. Corpus-based semantic class mining: Distributional vs. pattern-based approaches. In *COLING*. http://aclweb.org/anthology/C10-1112.

Chi Wang, Kaushik Chakrabarti, Tao Cheng, and Surajit Chaudhuri. 2012. Targeted disambiguation of ad-hoc, homogeneous sets of named entities. In *WWW*. https://doi.org/10.1145/2187836.2187934.

Jian Xu, Qin Lu, Jie Liu, and Ruifeng Xu. 2012. NLP-comp in TAC 2012 entity linking and slot-filling. In *TAC*. https://tac.nist.gov//publications/2012/papers.html.