A Probabilistic Model for Ranking Entities to Mentions

Shunlin Rong and Mizuho Iwaihara

Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikono, Wakamatu-ku, Kitakyushu-shi, Fukuoka-ken, 808-0135 Japan
E-mail: rongshunlin@ruri.waseda.jp, iwaihara@waseda.jp

Abstract

Entity linking (EL) is the task of mapping name mentions in web text to their entities in a knowledge base. Most of earlier EL work in the knowledge based approach is usually formulated as a ranking problem, either by (i) non-collective approaches with supervised models, or (ii) collective approaches by leveraging global topical coherence which means semantic relations between entities through graph-based approaches. For the mapping process, we can regard it as selecting an entity to its mention by combining these two methods. In this paper, we propose a probabilistic model that ranks related entities to name mentions where ranking is customized by using three types of data: popularity knowledge of the entity, context similarity between mentions and the entity, and semantic relations between mapping entities. Specifically, we first propose an EL model utilizing global topical coherence that means semantic relations between entities, as well as using local mention-to-entity compatibility, to improve recall and precision. The key benefit of our model comes from 1) combination of two methods to provide customized ranking for mentions, 2) the model is efficient by directly modeling the global semantic coherence between the entities and the document.

Keywords: Wikipedia, Entity Linking, Probabilistic Recommendation

1 Introduction

With the rapid increase of web data, the web has become one of the largest data repositories in the world in recent years. A large volume of data is stored on the Internet in the form of natural language texts which are often ambiguous. A named entity may have multiple names and a name could denote several different named entities.

The development of knowledge bases such as Wikipedia provide a possible solution to solve these problems. These knowledge bases contain rich knowledge about the world's entities, their semantic properties, and the semantic relations between each other. Take Wikipedia as an example, its 2016 latest English version contains more than 5 million articles which also refers to entities and 40 million article links between them. These knowledge base contain structured and unambiguous entities and entities relations. By mapping the unstructured web data to the structured and unambiguous entities, we can solve the ambiguous problem of web data.

A key problem in mapping web data to a knowledge base is linking name mentions in a document with their referent entities in a knowledge base which we often call the Entity Linking (EL) problem. For the mapping process, we could regard it as recommending an entity to a mention. The main difficulty of the task is as follows. The first problem is the name variation

problem which means an entity can be mentioned in different ways such as full names, aliases name and so on. The second problem is the name ambiguity problem, such as Michael Jordan could be referred as a famous NBA basketball player or a Berkeley professor.

Let us formulate the problem firstly. Let M = $\{m_1, m_2, \dots, m_k\}$ denote a collection of name mentions. Each name mention *m* in M is characterized by its name *m.s*, its local surrounding context *m.c* and the document containing it *m.d*. Given a knowledge base KB containing a set of entities E=$\{e_1, e_2, \dots, e_m\}$, the objective of our work is to recommend referent entities in KB of the name mentions in *M*. Let *m.e* denote the referent entity of a mention *m* in M. Our aim is to recommend a suitable entity *e* to a mention *m* by using a ranking method.

There are also several basic assumptions in this paper. The first assumption is that the more popular an entity is, the more likely it could appear in a document or paragraph. The second basic assumption is that the referent entity of a name mention should be topic coherent with its unambiguous contextual entities. To be detailed, the referent entities in one document should be semantic related. The third assumption is that the mentions and the mentions' referent entities should be similar in some extent, for example, the semantic similarities between the mentions and the mention's referent entities should be high.

For the first assumption, we consider that the popularity information of an entity tells us the likelihood of an entity appearing in a document. What's more the popularity of a mention to an entity may suggest the possible candidate entity for an ambiguous mention. For the second assumption, we consider that the referent entities $\{e_1, e_2, \dots, e_n\}$ of a set of mentions *M* in a document should be topic coherent. This uses the global semantic relations between entities in a document. For the third assumption, we use the local contextual similarities between name mentions and entities.

The main contribution of our work is as follows:
1) By modeling the global semantic relations between entities in a document directly, we avoid the traditional iterate methods which iterate many times to obtain an ideal result. We model the global semantic relation through a one-pass method which could be very efficient and obtain competitive results as these iteration methods in measuring the global the semantic relation between entities in one document.
2) We are the first trying to solve the problem in the view of recommendation. By applying this, we find the result is acceptable which means methods used in a recommendation system could also be used in entity linking decisions. Furthermore, our model works better as the corpus becomes larger, according to our experiments.

This paper is organized as follows. The related work is described in Section 2. The recommendation model is described in Section 3. Experiment results are presented and discussed in Section 4. Finally we conclude this paper in Section 5.

2. Related Work

Most of earlier entity linking problems can be formed as a ranking problem, either by (i) non-collective approaches which could be divided into two different parts.

The first is **local compatibility based approach** which is also the initial method by extracting the discriminative features of an entity from its textual description, then linking a name mention to the entity which has the highest contextual similarity with it. Mihalcea et al. [1] proposed a bag of words (BoW)-based methods, where the compatibility between a name

mention and an entity was measured as the cosine similarity between them. Cucerzan et al. [2] and Bunescu et al.[3] extended the BoW model by incorporating more entity knowledge such as entities' categories. One of its largest problems is that the dimension of vectors of the words sometimes becomes too big to calculate. Also they do not take into account the interdependence between EL decisions.

The second is **Simple Relational Approaches:** Considering the entity linking decisions in one document have no influence with each other, we can utilize the semantic relations between different entities in one document for linking decision. The core assumption is that the referent entity of a name mention should have a strong semantic relationship with its unambiguous contextual entities (Medelyan et al.,[4]). The main problem of this method is that they can only exploit pairwise interdependence between a name mention and its unambiguous contextual entities.

These methods deal with one mention at each time relying on prior popularity, context similarity, and other local features with supervised models without taking account of the global semantic relations between entities.

The second part is the collective methods which deal with the related mentions in parallel by leveraging the global semantic relationship between entities through graph-based approaches (X. Han et al[5]). This model needs to model the global semantic relations by iterate methods in one document which is not as efficient as the first method. On the other hand, it could achieve higher accuracy in entity linking decisions.

3. A Probabilistic Recommendation Model

In this section, we propose a probabilistic recommendation model for linking an entity to a mention. As far as we know, the closest work to us is Xianpei et al.[5] and Le et al.[6]. In the following sections we will introduce how to capture the popularity of entities in a knowledge base, how to calculate the local mention-to-entity compatibility, and how to measure the global semantic relation in a document.

3.1 Popularity of Entities

The reason why we want to capture popularity of entities in a knowledge base is to utilize the popularities for selecting candidate entities. The more popular an entity is, the more likely it could be a referent entity of a mention and appear in a document. As tried in the literature, we use the frequencies of an entity in the whole knowledge base to estimate the popularity of this entity. In this paper, we account the Wikipedia's redirect links which contain entities as the frequencies. Sometimes one redirect link may contain several entities which share same meanings. In this case, we choose the first one appears in this redirect link and account it appearing one time. This formula can be defined as follows:

$$\mathrm{P(e)} = \frac{Count(e) + 1}{|M| + N}$$

Where *Count*(*e*) is the count of the name mentions whose referent entity is *e*, and the $|M|$ is the total name mention count. The estimation is further smoothed by using the add-one method. Parameter N is the number of entities appearing in the whole Wiki dump.

As we can see this function needs a large number of documents to capture similarities. As the size of the knowledge base becomes larger, the model could measure the popularity better.

3.2 Semantic Relation between Entities

As we have mentioned in Section 1, the interdependency between two decisions on entity linking in one document means that they have semantic relations with each other. In other words, the referent entities of the name mentions in one document should be semantically related.

According to this observation, we utilize the Normalized Google Distance (Rudi et al[7]) to measure the semantic relatedness between two entities. However, the most difficult problem in this approach is that we cannot obtain referent entities in the other mentions when we are processing one name mention at a one-time pass. For example, basketball player Jordan plays for the Bull. When we deal with name Jordan, the candidate entities of Jordan would include Michael .I. Jordan who is a professor at Berkeley, and Michael Jeffrey Jordan who is a very famous basketball player. There are also other candidate entities for the Bull. It is not simple to calculate the semantic relatedness between the candidate entities of Jordan and Bull.

Existing researches [5] use graph-based methods to obtain the other name mentions' entities iteratively. It may iterate many times and sometimes cannot get an ideal result. Actually, the referent entity of the name mention should also have a semantic relation with the other remaining name mentions in this same document. In this example, the suitable referent entity Michael Jeffrey Jordan who is a famous basketball player of Jordan should also have a strong semantic relationship with the Bull itself, although the name Bull can have many meanings. Furthermore, the entity Michael I Jordan is not related to Bull. We use the Normalized Google Distance (NGD) to measure the semantic relatedness between the referent entities of one name mention and the associated name mentions in the document. The Google Distance is defined as follows:

$$NGD(x,y) = \frac{\max\{logf(x), logf(y)\} - logf(x,y)}{logM - min\{logf(x), logf(y)\}}$$

Where M is the total number of web pages searched by Google, $f(x)$ and $f(y)$ are the number of hits for search terms x and y, respectively, and $f(x, y)$ is the number of web pages on which both x and y co-occur.

Due to the limit on the Google API, we use another search engine Bing to calculate NGD. Commercial search engines are the most powerful web crawlers in the world. They crawl millions of new web pages on the Internet. By using a search engine to calculate NGD, we can deal with new words not in the knowledge base. On the other hand, the hit numbers of a search engine is a mixture on search results of non-disambiguated mentions. A commercial search engine will display all related pages about the name mentions, such as the pages about the Bull. Commercial search engines will display pages mixed about Chicago Bulls and pages about animals named bull, and so on.

The Normalized Google Distance measures the semantic relation between entities and the remaining name mentions. Actually, what we need is to measure the global semantic relationship in one document. If one candidate entity shares a stronger semantic relation with the other name mention in the same document, the entity has a higher possibility to be linked to the name mention. We address the situation using the following function:

$$\mathrm{Sd}(e_j) = \sum_{m \in d} (1 - NGD(e_j, m)) \quad m_j \neq m$$

In this formula, d means the document what we will deal with at a time. e_j is one of the candidate entities of name mention m_j. Also m is the name mentions except m_j in this document d. $Sd(e_j)$ measures the relationship between one entity e_j and the document d.

3.3 Local Context Similarities between Mentions and Entities

In this section we discuss capturing similarities between the local context of a mention m and a specific entity e. Traditional methods model this as a bag of words and map to a vector space where the name mention m is represented as a vector of its context words and entity e is represented as a vector of its Wikipedia words.

All words are weighted using their TFIDF values. This method sometimes consumes a great amount of time, since the dimensions of entity vectors can be thousands or even larger. To deal with this, we utilize word2vec by Tomas et al.[8] and Ilya et al. [9].

Word2Vec is an open source project released by Google which achieved state-of-the-art performances in many natural language processing tasks. It takes a large text corpus as input and outputs word vectors for each unique word. The resulting word vector file can be used as features in a number of natural language processing and machine learning applications.

By using outputs in the vector representation by word2vec, we can calculate the cosine similarity between a mention *m* and an entity *e*.

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

Here, the name mention *m* is represented as a vector of its context words and the entity *e* is represented as a vector of its Wikipedia page's words.

For candidate entity selection, we collect all redirect links as the anchor dictionary and count how many times it appears in the whole Wikipedia articles. Once we deal with the mention, we choose the similar one in the anchor dictionary. To be detailed, we use the wildcard to find all similar entities to this mention and filter it according to their popularities. For example, sometimes there are too many candidate entities for one mention. We will rank the candidate entities by their counts in descending order and choose top-10 as the mention's candidate entities.

3.4 Aim of the Model

In this section, we will describe the aim of our model as follows:

$$\text{rank}(m,e) = P(e) * CP(m,e) * Sd(e)$$

Here, *m.e* is the referent entity of the name mention *m*. *P(e)* measures the popularity of the entities *e* and *CP(m,e)* measures the local similarities between name mentions and entities. As defined earlier, *Sd(e)* measures the relatedness between document *d* and entity *e*. A most likely entity *e* is then determined by these three factors.

4 Experiments

In this section, we evaluate the performance of our method and compare it with the traditional methods. We first explain the experiment settings in Section 4.1, then we will discuss and evaluate the results in Section 4.2.

4.1 Experimental Setting

4.1.1 Knowledge Base

We use the Dec.20, 2015 English version of Wikipedia as the knowledge base. In total, the knowledge base contains more than 9,000,000 distinct entities. A name-to-entity dictionary contains over 17,000,000 distinct entity names and the candidate referent entities of each name. There are over 400,000,000 semantic relations between entities.

As to the measure methods, we use the sax to deal with the whole wiki dump and we treat one article as one entity. Once we deal with the whole wiki dump, we can get the number of the articles in the dump which means we can get the number of entities in the whole dump. We also treat the redirect words in the wiki dump as names of the entities and regard links between redirect words and the document which contains these redirect words as kind of semantic relations.

4.1.2 Dataset

Our experiments are composed of three parts. The first part is to calculate the semantic relatedness between entities in one document. We need to use the hit numbers of a commercial search engine, and we use Bing. The second part is to incorporate Wikipedia articles as the ground truth of the entity linking results. To be detailed,

we use the contents and redirect links of the Wikipedia as the ground truth. The third part is the corpus we use to evaluate the effects of our model. In this paper, we use the KORE dataset which is used in [11] containing a large number of very ambiguous mentions from five domains (Celebration, Musician, Business, Sports, Politics).

To evaluate the effects of our model, we first test it on a relative small wiki dump which contains 19105 articles and 4015004 entity relationships and then test on a larger wiki dump which contains 9435689 articles and 419250479 entity relationships, to test our assumption that the model performs better on a large data set. Then we will compare our model with the baseline. For the input documents, we choose several small paragraphs.

4.1.3 Evaluation Criteria

As this is a recommendation model, what we concern most is the precision and recall between the referent entity and mention *m*. The measures are given as follows:

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

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

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Here, *M* is the set of referent entities produced by our entity linking method, *M** is the golden standard set of the referent entities judged by human. In this experiment, I manually checked ranked results and count entities. So there is a risk that the result would be influenced by my tastes. I would invite several guys to do these the judges together to avoid this problem.

4.1.4 The Baseline

TagMe (Ferragina and Scaiella, 2010) is an end-to-end Wikification system specialized in short texts. TagMe performs best among publicly available Wikification systems. We adopt the TagMe as the baseline to judge our model's performance on short texts. The human judge is done by us according to our common senses.

4.2 Experiment results

We conduct experiments on small corpus first to check the result of our model on short texts. Actually, for short texts as we experiment below，it is not useful to model the local similarities between name mentions and entities, because the numbers of name mentions in short texts are too small to be compared with entities. In this situation of short texts, we will use our simplified model by only considering the popularities of entities and the global semantic relatedness between entities and name mentions. The detailed result is shown as follows:

Domain	Mentions	Candidate Entities	word count
Celebration	50	897	119
Musician	68	1155	191
Business	70	1350	183
Sports	30	540	76
Politics	41	811	123

Table 1: the facts about two documents

We list the number of words, the number of mentions and the number of candidate entities for the fifty documents in Table 1.Every domain has 10 documents. To improve recall, we choose the top 20 candidate entities for every mention. Actually not all name mentions have 20 candidate entities. The word count is the number of words in each domain.

In this experiment, we also take use of the Standford University tools postag[12] to preprocess the text to find the name mentions in texts.

The experiment results of TagMe, our model PMRE with whole wiki dump PMRE(B) and part of wiki dump PMER(S) are shown as follows:

Domain	TagMe	PMRE(S)	PMRE(B)
Celebration	0.3333	0.3143	0.5750
Musician	0.6805	0.4400	0.6863
Business	0.5079	0.2962	0.4590
Sports	0.4516	0.1739	0.3333
Politics	0.4651	0.7500	0.6216
Total	0.5077	0.4615	0.5462

Table 2: Precision values of two models

Domain	TagMe	PMRE(S)	PMRE(B)
Celebration	0.3400	0.2200	0.4600
Musician	0.7205	0.3235	0.5147
Business	0.4571	0.2286	0.4000
Sports	0.4666	0.1333	0.3000
Politics	0.4878	0.6585	0.5610
Total	0.5097	0.3707	0.4556

Table 3: Recall values of two models

Domain	TagMe	PMRE(S)	PMRE(B)
Celebration	0.3366	0.2588	0.5111
Musician	0.7000	0.3729	0.5882
Business	0.4812	0.2581	0.4274
Sports	0.4590	0.1509	0.3157
Politics	0.4761	0.7013	0.5897
Total	0.5087	0.4111	0.4968

Table 4: F1 values of two models

According to the experiments, we find that the model with whole wiki dump PMRE(B) performs better than the model with part wiki dump PMRE(S) in precision values, recall values, F1 values proving our assumption that the model will perform better on larger document sets.

Comparing the model PMRE(B) with TagMe, we find that the model PMRE(B) performs better than then the TagMe in Celebration and Politics parts while the TagMe performs better in the three remaining parts in these values. In total we can see that the TagMe performs a litter better than the PMRE(B) and far better than PMRE(S).It is remarkable that our model is wining in several domains. It performs better on Musician domain, Politics domain and performs worse on sports domain. The TagMe performs better on Musician and worse on Celebration and Sports domain. The first reason is the sports corpus are too short compared to other domain corpus. It is not much useful to take use of the context words. The second reason is that the popularity of the persons appearing in this domain could not compete with the popularity of the person appears in the Musician domain and sports domain. Candidate entities for the domain is not that suitable.

As TagMe performs best among publicly available Wikification systems on short texts. We can see that our system can compete with the state of the art system in many cases. We expect that an even larger dataset can improve our result.

5 Conclusions and Future Work

In this paper, we proposed a probabilistic model for ranking entities to mentions by utilizing global topical coherence that means semantic relations between entities, as well as using local mention-to-entity compatibility, to improve recall and precision. According to our experiments on short texts, our model could compete with the TagMe. In the future, we will perform this model on larger corpus and longer documents. What's more, according to our assumption, our model should be efficient, we will do more experiments to confirm this.

6 REFERENCES

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

[2] Cucerzan, S. 2007. Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of EMNLPCoNLL.

[3] Bunescu, R. & Pasca, M. 2006. Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of EACL, vol.6.
[4] Medelyan, O., Witten, I. H. & Milne, D. 2008. Topic indexing with Wikipedia. In: Proceedings of the AAAI WikiAI workshop.
[5] X. Han, L. Sun, and J. Zhao. 2011. Collective entity linking in web text: A graph-based method. In Proc.SIGIR2011.
[6] X.Han, L.Sun.2011. A Generative Entity-Mention Model for Linking Entities with knowledge Base in Proc. ACL2011
[7] Rudi L. Cilibrasi and Paul M.B. Vita ńy. 2007. In IEEE Transactions on knowledge and data engineering.
[8] Strube, M. and Ponzetto, S. P. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In: Proceedings of AAAI.
[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.
[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013
[11] Johannes Hoffart., CIKM 2012 KORE: Keyphrase Overlap Relatedness for Entity Disambiguation
[12] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, et al 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.