Evaluating Semantic Relatedness through Categorical and Contextual Information for Entity Disambiguation

Yiming Zhang[†] and Mizuho Iwaihara[‡]

早稲田大学大学院情報生産システム研究科

〒808-0135 福岡県北九州市若松区ひびきの２－７

E-mail: †zhang_ym@toki.waseda.jp, ‡iwaihara@waseda.jp

Abstract The number of entities in large-scale knowledge bases has been growing in recent years. The key issue to entity linking using a knowledge base such as Wikipedia is entity disambiguation. The objective of our proposing system is to disambiguate entities in documents and link entity mentions to their corresponding Wikipedia articles. To this end, our system ranks the set of candidate entities based on relatedness by utilizing semantic features derived from Wikipedia category hierarchies and articles. In addition, to reflect contextual information of Wikipedia, we utilize word embedding for refining the ranking result of candidate entities. Our experiment results show that these features has given good correlation with human rankings in candidate relatedness ranking and also high disambiguation accuracy on news articles.

Keyword Entity Disambiguation, Semantic Relatedness, Entity Similarity

1. Introduction

1.1. Entity disambiguation

With the expansion of web, more and more text data can be approached easily. But looking through enormous text data might distract people's attention. It is necessary to extract useful and refined information from mass text. To solve this problem, researchers have been studying related techniques, such as document summarization, text classification, and text clustering. Meanwhile, entity-centric data has led researchers to a new direction of constructing an entity based network, and Google Knowledge Graph is a well-known example. Changes are taking place in web text data, from keyword-based text organization to an entity-centric network of knowledge. An eminent feature is to replace the outline of documents described by simple keywords with a deeper relationship between entities mentioned in documents.

Web pages, twitter, blog postings and news articles contain mentions of entities, such as people, organization and geo-political places. To help readers easily understand the meaning of main concepts and decrease reading difficulties, those mentions are recommended to link to a corresponding descriptive entity page. Sometimes mentions are ambiguous, a name may refer to different entities in different contexts. For example, a mention of *Washington*, can be linked to dozens of possible entities in Wikipedia, such as *George Washington*, *Washington, D.C.*, *University of Washington*. So this ambiguity problem will hinder the accuracy of entity linking. As another example, given a sentence: "*Texas is a pop rock band from Scotland and its name was from the 1984 Wim Wenders' movie Paris, Texas*". But how can a computer distinguish that *Texas* is a music band, not a state in the United States and that "*Paris, Texas*" denotes a drama film rather than a city in Northeast Texas or another band called Paris, Texas?

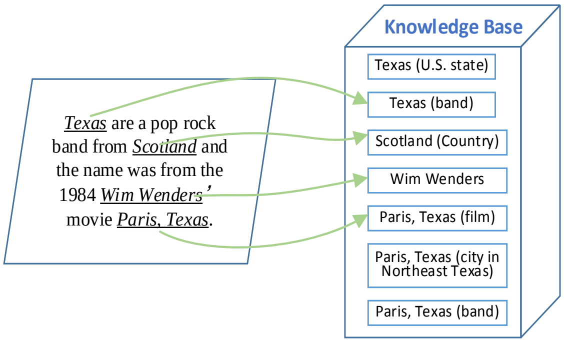

Figure 1 Illustration of entity disambiguation task

So the aim of entity disambiguation is to solve this problem, establishing mappings between the mentions and the actual entities from a knowledge base, like the example in Figure 1, showing the final linking result.

1.2. Wikipedia and its structure

Wikipedia is a multilingual free internet encyclopedia, which is collaboratively edited by volunteers worldwide whose efforts have resulted into over thirty million articles.

There are more than five million articles in English Wikipedia. Because of its abundant coverage and relative up-to-date resources, Wikipedia is often used as a knowledge base for entity disambiguation. So from this sense, one Wikipedia page is regarded as an entity, the page title is regarded as an entity name, and page content is as entity description or context. By doing this, we can link a mention to a Wikipedia page.

An especially interesting aspect of Wikipedia structure is the categorization and linkage within its content [1][8]. Categories in Wikipedia have a hierarchical structure (Figure 2), there is a relative root category called *Main topic classification*, which can be backtracked by each bottom category. Each category can have an arbitrary number of subcategories, and similarly, one category may have more than one parent categories, while cycles and disconnected categories are possible. One page is explicitly assigned to one or more categories during users' editing. Categories act as a semantic tag, and articles that have similar topics usually belong to the same category. There also exist numerous links between categories and pages, from categories to sub-categories and from categories to pages.

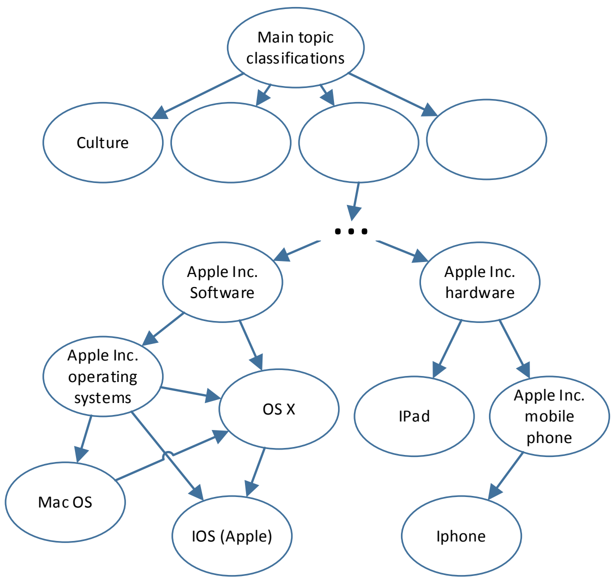

Figure 2 Wikipedia category organization

1.3. Word embedding

Distributed representations for words were proposed early, but recently have been successfully used in language models and some natural language processing tasks, including word embedding learning [7][12]. Traditional word representations are based on a co-occurrence matrix of size $W \times C$, where W is the vocabulary size, each row of this matrix is the representation of word w, and each column is the word's context but suffers from high dimensionality. Recent word embedding has brought new ideas to this area.

1.4. Our contribution

Our work exploits Wikipedia category hierarchical features, word embedding similarity and Wikipedia in-link relations for measuring semantic relatedness between words, and these bases for semantic relatedness contribute to achieve the entity disambiguation goal. The rest of the paper is organized as follows: Section 2 gives an overview of various approaches to entity disambiguation and related work on word embedding. Section 3 describes our system methods and features. Section 4 presents an experiment for evaluating sematic relatedness, comparing the effects of different features, and made an analysis of the results. The paper concludes in Section 5.

2. Related work

The usage of Wikipedia has provided tremendous chances for natural language processing, with various derived huge knowledge bases, such as DBpeida, Freebase, and BabelNet. Text processing tasks including entity linking have benefited greatly from these resources. The problem of named entity disambiguation was addressed in the literature from different perspectives. Using link based structure and using text in articles are two main Wikipedia-based approaches.

Milne and Witten [2] researched semantic relatedness in the early time. They are the first to use hyperlink between articles for semantic relatedness measurement. They compared two Wikipedia pages by computing the number of incoming links the two pages have in common based on Normalized Google Distance (NGD). But this sort of transformation of NGD ignores out-links and links that should appear but missed or ignored during Wikipedia editing. In later work [3], Milne and Witten incorporated machine learning, applying their relatedness measurement to train a supervised classifier in entity disambiguation. Their method has shown effectiveness to some extent. However, these approaches do not consider joint dependencies among the possible target entities from input context.

A distinct direction in entity disambiguation focuses on the effects of mention's context. The referent entities of a mention is reflected by its context. Context similarity measures how similar the text around the named entity in the text data and the text in the Wikipedia article are. TF-IDF measure and cosine similarity function are often used. TagMe [4] uses a light-weight form of context coherence by averaging relatedness scores over all candidate entities for a mention that co-occurs with a given mention.

As for word embedding, the main advantage of word embedding is that the word representation of two similar words are very close in vector space and the dimension of words can be decreased greatly in comparison with the traditional bag-of-word model. Word2Vec [1] is an open source project released by Google which achieves state of the art performances. It takes a large text corpus as input and outputs word vectors for distinct words. These word vectors can be subsequently used in various natural language processing and machine learning applications. In Mikolov et al.'s Word2Vec paper [7], they carried out two neural network models for representation learning: continuous bag-of-words model (CBOW) and the continuous skip-gram model. It was shown that word representations can capture semantic information between words.

There are other researches on the analysis of Wikipedia's category hierarchies. In [8], authors performed a graph-theoretic analysis of the Wikipedia Category Graph and showed that the Wikipedia category graph is well suited to estimate semantic relatedness between words.

There are also a number of systems presenting complex combinations of several methods for entity disambiguation. Lemahnn et al. [9] present a supervised system combining features based on hyperlinks, categories, text similarity and relations from info-boxes.

Our research is based on former research and propose new methods to make use of these features.

3. Proposed method

3.1. Mention recognition

The entity disambiguation process is usually divided into two steps: recognition and disambiguation. In the first step, the system recognizes potential mentions from the input text and links them to a set of candidate entities which are likely to be referred to by the recognized mention.

In some existing entity linking approaches, researchers usually use an existing named entity recognition (NER) tagger to implement the entity recognition. One popular tool is Stanford NER tagger which used conditional random fields (CRF) Classifier, and it can recognize particularly three classes, i.e. person, organization, and location. So firstly we use NER tagger, to obtain person, location and organization phrases. Then our system will compare these phrases with Wikipedia titles based on the following rules:

- For a phrase that is an exact match with Wikipedia page title: If the Wikipedia page is not a redirect page nor a disambiguation page, then this phrase is regarded as an unambiguous entity and will be linked to this Wikipedia page finally.
- For a phrase that is contained in or contain a Wikipedia page title (e.g. noun *apple* and title *Apple Inc.*): Then the candidate set will include all of these possible candidate titles.
- For a phrase that is redirect or disambiguation page title of Wikipedia: The redirect page and disambiguation page will be retrieved, and append all out-link page titles to the candidate set.
- For a phrase that is an abbreviation of Wikipedia titles, these possible titles will be added into the set of candidate entities.

As this part of research is still in process, for now we can just obtain a rough set of candidate entities. Because certain phrases are likely to produce a large candidate set, so pruning work should be done in future in order to reduce the computation cost of following disambiguation part.

3.2. Entity disambiguation

After the first step, we can obtain a set of candidate entities for each mention in the input text. Then in the second step, for every mention its candidate entities need to be ranked by semantic similarity and other approaches. The candidate that obtained the highest score will be chosen as the final entity the mention referred to.

3.2.1. Wikipedia superordinate category based feature

Categories in Wikipedia have a hierarchical structure, where every article is assigned to one or more categories when editor is editing it. Each category can have an arbitrary number of subcategories, and may have more than one parent categories. Pages which have common parent categories usually have high semantic relatedness. Most of page-category links provide navigation within Wikipedia contents, furthermore, they also inferred some semantic relationships between pages.

Let us see an example in Figure 3, depicting page *Apple Inc.* having several categories, e.g. *Apple Inc.*, *Computer*

[1] https://code.google.com/p/word2vec/

hardware companies. For category *Apple Inc.*, there are tens of subcategories (such as category *Apple Inc. hardware* and category *Apple Inc. advertising*) and pages (such as *Steve Jobs*) under it. For category Computer hardware companies, there exist dozens of subcategories (such as *Motherboard companies*) and pages (such as *Microsoft*) under it. Furthermore, subcategory *Apple Inc. hardware* also has its subcategories (such as *IPad*) and pages under it. In this way, super-categories and subcategories construct the category hierarchy. From experience, we know that the semantic relationship between *IPad* and *Steve Jobs* is stronger than *IPad* and *Microsoft*, and from this category diagram, we can see that *Steve Jobs* and *IPad* are under the same parent category.

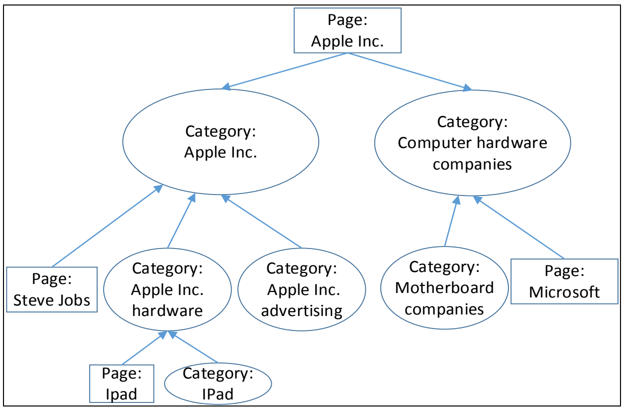

Figure 3 Wikipedia category example

As described in the above, one of our features exploits category hierarchies. The categories we use are extracted from Wikipedia dump on Feb. 5th, 2015. We clean the category hierarchy, by removing Wikipedia management categories, words like Wikipedia, mediaWiki, wikiproject, template are included. Then we construct a category set as described in our algorithm below.

```
for each mention in input:
    if candidate_set.size()==1:
        comparison_set.add(candidate.getAscendantCategory())
for each mention in input:
    if candidate_set.size()>1:
        for cand in mention.candidate_set:
            cand.categ_set=cand.getAscendantCategory()
            cand.categ_score=J(categ_set, comparison_set)
```

The system will then scan every mention that has been tagged in the recognition part. The first scan only scans unambiguous mentions, aiming to construct a common comparison set (*comparison_set*) for further use. When there is only one candidate in the mention's candidate set, then this mention is unambiguous, so the system will fetch this entity's all ascendant categories (by *getAscendantCategory()*) and store them in a common comparison set. To achieve a faster retrieval and remove too many overlappings between superordinate categories, the depth range is set between 3 and 6. When we set the depth of recursion to 3, that is to store all the categories of this entity and the parents and grandparents of the categories. A higher depth of recursion is time-consuming and will not deliver a prominent improvement. Later the system will scan all ambiguous mentions and compute category scores for each candidate. If there are more than one candidates in the set of mention's candidate set, an iterative calculation of fetching ascendant categories (*cand.getAscendantCategory()*) for every candidate of this mention will be done. Then the system will compute a category score (*categ_score*) using this candidate's ascendant categories (*categ_set*) and the common comparison category set (*comparison_set*) produced by the first scan. Here our category score calculation uses the Jaccard similarity function:

$$Jaccard(comparison\_set, \text{candidate\_}categ\_set) = \frac{|\ comparison\_set \cap \text{candidate\_}categ\_set\ |}{|\ comparison\_set \cup \text{candidate\_}categ\_set\ |}$$

The higher the category score is, the more similar between the candidate and those unambiguous entities from input text is. Chances are that this candidate has some relationship with other entities from the input. So there is a high possibility of similar entities co-occurring in context.

3.2.2. Embedding word similarity

Word embedding is a dense and low-dimensional word representation. Each dimension of the embedding represents a latent feature of the word, capturing useful syntactic and semantic properties. We decide to use Word2Vec model [7]. The Word2Vec algorithm is used for constructing a word embedding for each unique word in the corpus. We adopt the hierarchical softmax skip-gram model which uses 2N surrounding words to predict the middle word and builds up a Huffman tree to represent probability distribution of words in the corpus. The vector for each word is a semantic description of how that word occurs in context. So if two words are usually used similarly in given corpus, chances are that they will obtain similar representations. After mapping words into the vector space, we can compute cosine similarity to find words that have similar semantics. The trained word embeddings largely depend on textual content, that is the training corpus. In order to capture more semantic information, we use the whole Wikipedia corpus as training corpus in the experiment. We use Gensim [1] under Python to load prebuilt model [2].

The similarity in word2vec is defined by the cosine similarity of two word vectors. So we will also use this similarity measurement. For a sentence in the user's input, its similarity with the candidate is calculated by computing the mean word vector similarity between these words and the candidate entity. We compute the following score on entities.

$$\text{score}(input, mention\_candidate) = \text{sim}\left(\frac{\sum_{input}^{m} V_m}{|input|}, \frac{\sum_{mention\_candidate}^{w} V_w}{|mention\_candidate|}\right)$$

input is user's input query or context of a mention in the processed article (mention is tagged by recognition step). *m* is every word in *input*. $|input|$ is the number of words in *input*. If the article is too long, we set the context window to 100 words around the mention. *mention_candidate* is the preprocessed Wikipedia article for one candidate from the mention's candidate set. *w* is every word in this candidate name. $|mention\_candidate|$ is the number of words in this candidate name.

3.2.3. Wikipedia inlink-based feature

Wikipedia inlinks are links from one Wikipedia page to another Wikipedia page. Inlinks are added during user's editing. We suppose that if two Wikipedia pages have more inlinks in common, the more pages are related to both pages, so we can say they are highly correlated. Our link relatedness is defined as the conditional probability of entity c given entity e [5][12]:

$$\text{InR}(c|e) = \frac{|In(c) \cap In(e)|}{|In(e)|}$$

4. Experiment and Evaluation

4.1. Entity relatedness experiment

4.1.1. Dataset

We implemented our proposed system based on Wikipedia knowledge base, using the snapshot of English Wikipedia dump exported on February 5th, 2015 [3].

In order to compare the quality of relatedness scores between entities obtained by our approach, we performed experiment on entity relatedness. In the KORE-relatedness-entity dataset [10], the authors selected a set of 21 queries which correspond to 21 entities from knowledge base YAGO. These entities are selected from four different domains: IT companies, Hollywood celebrities, video games, and television series (Table 1). For each of the 21 seed entities, they selected 20 candidates from the set of entities linked to the seed's Wikipedia article. The authors used a crowdsourcing platform to obtain the gold standard ranking of the 20 candidate entities for each seed entity. The KORE dataset is composed of 420 entity pairs in total.

We compare our ranking result with the gold standard of human-ranked results and evaluate our method using Spearman's rank correlation coefficient (ρ). Spearman's correlation assesses how well the relationship between two variables can be. Spearman correlation ranges from -1 to 1, a perfect positive correlation is represented by the value 1, while a value 0 indicates no correlation and -1 indicates a complete negative correlation. From this sense, the correlation ρ of our system's result and gold-standard ranks should be a positive value, and the larger is the better.

Seed entity	Ranked Candidate Entities
Apple Inc. (IT Companies)	Steve Jobs (1), Steve Wozniak (2) Jonathan Ive (3), Mac Pro (4) … Greenpeace (17), Ginza (18) Sears (19), Ford Motor Company (20)
Angelina Jolie (Hollywood Celebrities)	Jon Voight (1), Brad Pitt (2) … Chip Taylor (8), Academy Awards(9) … 2005 Kashmir earthquake (20)
Grand Theft Auto IV (Video Games)	Niko Bellic (1) Grand Theft Auto (series) (2) … Brooklyn Bridge(14), Metacritic (15) …
Mad Men (TV Series)	Matthew Weiner(1), Jon Hamm(2) Alan Taylor (director) (3) … Volkswagen Beetle (19) Sesame Street (20)

Table 1 Example of seed entities and gold-standard ranks of candidate entities

4.1.2. Experiment result and analysis

Experimental results for the KORE relatedness entity dataset are shown in Table 3. The values in the table are the Spearman correlation between the gold-standard rank and the ranking result generated by our methods.

[1] https://radimrehurek.com/gensim/
[2] https://github.com/idio/wiki2vec
[3] https://dumps.wikimedia.org/

Domain	**SuC**	**WoE**	**InK**	$\boldsymbol{KORE_{LSH+F}}^{(\text{baseline})}$
IT companies	0.2815	**0.4722**	0.4478	0.208
Hollywood Celebrities	0.3982	**0.5546**	0.3702	0.522
Video Games	**0.5800**	0.4770	0.4710	0.499
Television Series	0.2017	0.1119	**0.3862**	**0.426**
Chunk Norris	0.2133	0.3789	**0.4992**	**0.653**
Average(21 entities)	0.3571	0.4027	**0.4226**	**0.425**

Table 3 Spearman correlation of relatedness measures with human ranking

The SuC method is the method based on superordinate category feature. The WoE method is the word embedding similarity method. The InK method is the method based on Wikipedia inlink feature. The KORE method in the last column is one of the methods proposed by the authors of the KORE dataset, and we regard it as our baseline.

In Table 3, we compared the results of rank correlation. The WoE method achieved the best scores of *IT companies* domain and *Hollywood Celebrities* domain. The best scores of *Television Series* domain and the separate entity *Chunk Norris* which do not belong to any domain are obtained by the baseline KORE method. Our methods outperform the baseline on three domains.

Gold standard rank	SuC rank	Related candidate	Num of common categories with entity
3	1	Grand Theft Auto III	845
2	2	Grand Theft Auto (series)	689
1	3	Niko Bellic	526
7	4	PlayStation 3	318
4	5	Rockstar Games	316
		…	
16	17	Edinburgh	171
11	18	New York City	146
18	19	Federal Bureau of Investigation	115
13	20	Eastern Europe	20

Table 2 SuC Rank comparison for *Grand Theft Auto IV*

Then we focus on the analysis of the feature based on Wikipedia superordinate category. It obtained best score for *Video Games* domain. We take the *Grand Theft Auto IV* entity as an example, as shown in Table 2. It gained a correlation score of 0.712, which is the highest score obtained by SuC. When we set ascendant category recursion depth to 4, we can obtain all the superordinate categories for *Grand Theft Auto IV*. The candidates ranked top have more common categories with the seed entity, in contrary, candidates that ranked lower have fewer common categories with the seed entity. But this is not always the truth, such as for seed entity *Apple Inc.* and *IBM*, there is little difference between the common categories among candidates. So the result is not good for these seed entities.

Lastly, we analyze the performance of the Word embedding method. It works best for *Hollywood Celebrities* domain. We take *Brad Pitt* as an example. Table 4 shows the rank comparison of the word embedding result and gold-standard, and the word2vec similarity between every candidate entity and the seed entity *Brad Pitt*. The name which is unique and has less duplication is easy to distinguish the person. In this example, Brat Pitt is a famous actor, and he and other famous actors, like *Rusty Ryan*, co-starred in some movies. So their names often co-occur in Wikipedia pages which can be movies' pages, directors' pages, and other co-starred actors' pages and so on. *Jennifer Aniston* is *Brad*'s ex-wife and *Angelina Jolie* is his current wife. So their names co-occur a number of times. Word2vec captured such co-occurrence and context information, making these candidates in the top rank. As for the latter candidates in this table, like CNN, Sudan, Pakistan, they seldom co-occur with Brad Pitt. So they ranked lower.

Gold standard rank	WoE rank	Related candidate	Word2vec similarity
1	1	Angelina Jolie	0.4715
4	2	Rusty Ryan	0.4217
2	3	Jennifer Aniston	0.3789
11	4	David Fincher	0.3703
		…	
17	17	Nice	0.0313
19	18	CNN	0.0208
18	19	Sudan	0.0185
20	20	Pakistan	-0.0189

Table 4 WoE Rank comparison for *Brad Pitt* entity

Word embedding similarity performed badly in TV Series domain. We take the worst one as an example. Table 5 shows the rank comparison result of entity *The Wire*. This entity's rank correlation obtained -0.44, which means negative correlation. We can analyze some reasons. Firstly the entity name contains a 'The', as we know 'The' is too common in Wikipedia, which impairs the word vector representation for this entity. Also, word2vec is good at comparing the similarity between separate words, but does not work well in comparing phrases on which word orders need to be considered.

Gold standard rank	WoE rank	Related candidate	Word2vec similarity
18	1	Chicago Sun-Times	0.2707
7	2	HBO	0.2684
19	3	Soul Food (TV series)	0.2592
17	4	Six Feet Under (TV series)	0.2493
...			
13	17	Bob Ehrlich	0.0899
11	18	Kurt Schmoke	0.0779
6	19	Idris Elba	0.0771
10	20	Blake Leyh	0.0720

Table 5 WoE Rank comparison for *The Wire* entity

4.2. Disambiguation experiment

4.2.1. Dataset

Original AQUAINT corpus consists of newswire text data in English, drawn from three sources: the Xinhua News Service, the New York Times Service and the Associated Press Worldstream News Service. We used a subset of AQUAINT corpus, manually removing repeated articles and mentions that are common words, and verified links to Wikipedia titles according to specific version. So our dataset contains 45 news articles and 370 mentions, among which 266 mentions have at least 2 candidates. There are about average 220 words and average 8 mentions in every news article.

For some mentions, there are too many candidates. We only keep the top 10 candidates ranked by their popularities in Wikipedia and if the true entity is not in the candidate set, we add the true entity to its candidate set.

4.2.2. Experiment result and analysis

Method	Correctly linked mention	Accuracy
SuC	257	69.5%
InK	244	65.9%
WoE	280	75.7%

Table 6 Disambiguation results on dataset

We measured the accuracy as the fraction of mentions that were correctly linked to Wikipedia pages. Our disambiguation experiment result is shown in Table 6. The SuC method is the method based on superordinate category feature. The InK method is the method based on Wikipedia inlink feature. The WoE method is the word embedding similarity method.

Overall, the WoE method performed the best, it can correctly link more than 75% mentions to its true entity in Wikipedia page and followed by the SuC method.

Mention	SuC	InK	WoE
Air Afrique	0	0	0
Boeing	1	1	0
Nigeria's largest international airport	0	0	0
Lagos	0	2	1
Guardian newspaper	1	0	1
Cameroon Airlines	0	0	0
Nigeria	0	1	1

Table 7 Ranking of true entity in the candidate set using 3 methods for one article

Let us take one article (APW19980625_1136) for an example to analyze the results. There are seven mentions in this article and two mentions (*Air Afrique* and *Cameroon Airlines*) are unambiguous. In Table 7, *0* means the true entity is correctly ranked at the first place, and ranking results other than 0 mean this method could not choose the correct entity for this mention.

5. Conclusion and future work

This paper proposed an entity disambiguation system based on the Wikipedia knowledge base. It is modeled to exploit the Wikipedia category hierarchies and Wikipedia links structure. Moreover, to measure the semantic relatedness, word embedding is used to refine the result. Based on semantic relatedness, we can obtain a rank of candidate entities for one mention. We also evaluated our methods on disambiguation experiment.

In future, we plan to add more context information to our method and learn a combination method based on machine learning algorithms.

Reference

[1] Bairi, Ramakrishna B., Mark Carman, and Ganesh Ramakrishnan. "On the Evolution of Wikipedia:

Dynamics of Categories and Articles." Ninth International AAAI Conference on Web and Social Media. 2015.

[2] Witten, I., & Milne, D. (2008, July). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA (pp. 25-30).

[3] Milne, D., & Witten, I. H. (2008, October). Learning to link with wikipedia. In Proceedings of the 17th ACM conference on Information and knowledge management (pp. 509-518). ACM.

[4] Ferragina, P., & Scaiella, U. (2010, October). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In Proceedings of the 19th ACM international conference on Information and knowledge management (pp. 1625-1628). ACM.

[5] Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R., & Trani, S. (2013, October). Learning relatedness measures for entity linking. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (pp. 139-148). ACM.

[6] Agirre, E., Barrena, A., & Soroa, A. (2015). Studying the Wikipedia Hyperlink Graph for Relatedness and Disambiguation. arXiv preprint arXiv:1503.01655.

[7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[8] Zesch, T., & Gurevych, I. (2007, April). Analysis of the Wikipedia category graph for NLP applications. In Proceedings of the TextGraphs-2 Workshop (NAACL-HLT 2007) (pp. 1-8).

[9] Lehmann, J., Monahan, S., Nezda, L., Jung, A., & Shi, Y. (2010). LCC approaches to knowledge base population at TAC 2010. In Proc. TAC 2010 Workshop.

[10] Hoffart, J., Seufert, S., Nguyen, D. B., Theobald, M., & Weikum, G. (2012, October). KORE: keyphrase overlap relatedness for entity disambiguation. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 545-554). ACM.

[11] Taieb, M. A. H., Aouicha, M. B., & Hamadou, A. B. (2013). Computing semantic relatedness using Wikipedia features. Knowledge-Based Systems, 50, 260-278.

[12] Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic Regularities in Continuous Space Word Representations. In HLT-NAACL (pp. 746-751).